Homework12_BACS

109090035 helped by 109070022,109060082,109070028

5/3/2023

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)

```
# load the car dataset
cars <- read.table("/Users/user/Downloads/auto-data.txt", header=FALSE, na.strings = "?")

names(cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "model_year", "origin")

# Remove unnecessary columns (displacement, horsepower, and NA)
cars <- subset(cars, select = -c(displacement, horsepower))

# Create cars_log data frame with log-transformed columns
cars_log <- cars
cars_log$mpg <- log(cars$mpg)
cars_log$weight <- log(cars$weight)
cars_log$acceleration <- log(cars$acceleration)

# Print cars_log data frame
head(cars_log)
```

```
       mpg cylinders   weight acceleration model_year origin
1 2.890372         8 8.161660     2.484907         70      1
2 2.708050         8 8.214194     2.442347         70      1
3 2.890372         8 8.142063     2.397895         70      1
4 2.772589         8 8.141190     2.484907         70      1
5 2.833213         8 8.145840     2.351375         70      1
6 2.708050         8 8.375860     2.302585         70      1
                         NA
1 chevrolet chevelle malibu
2         buick skylark 320
3        plymouth satellite
4             amc rebel sst
5               ford torino
6          ford galaxie 500
```

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

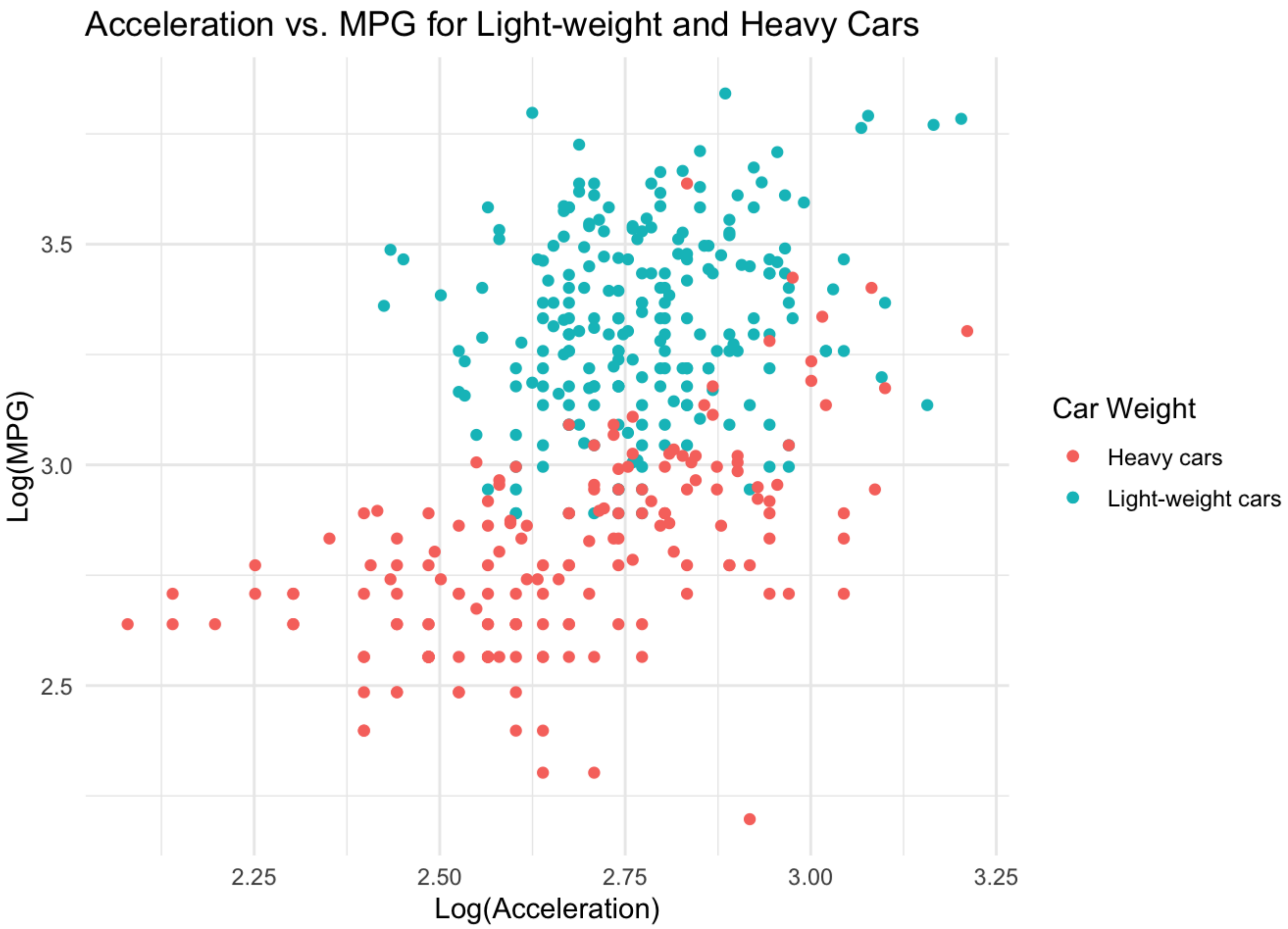
a. Let’s visualize how weight might moderate the relationship between acceleration and mpg:

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

ii. Create a single scatter plot of acceleration vs. mpg, with different colors and/or

shapes for light versus heavy cars

```
# Calculate the mean weight of the original cars dataset
mean_weight <- mean(cars$weight)

# Create two subsets of cars_log data, one for light-weight cars and one for heavy cars
light_cars <- subset(cars_log, weight < log(mean_weight))
heavy_cars <- subset(cars_log, weight >= log(mean_weight))


# Create a scatter plot of acceleration vs. mpg for light-weight and heavy cars
ggplot() +
  geom_point(data = light_cars, aes(x = acceleration, y = mpg, color = "Light-weight cars")) +
  geom_point(data = heavy_cars, aes(x = acceleration, y = mpg, color = "Heavy cars")) +
  labs(title = "Acceleration vs. MPG for Light-weight and Heavy Cars",
       x = "Log(Acceleration)",
       y = "Log(MPG)",
       color = "Car Weight") +
  theme_minimal()
```

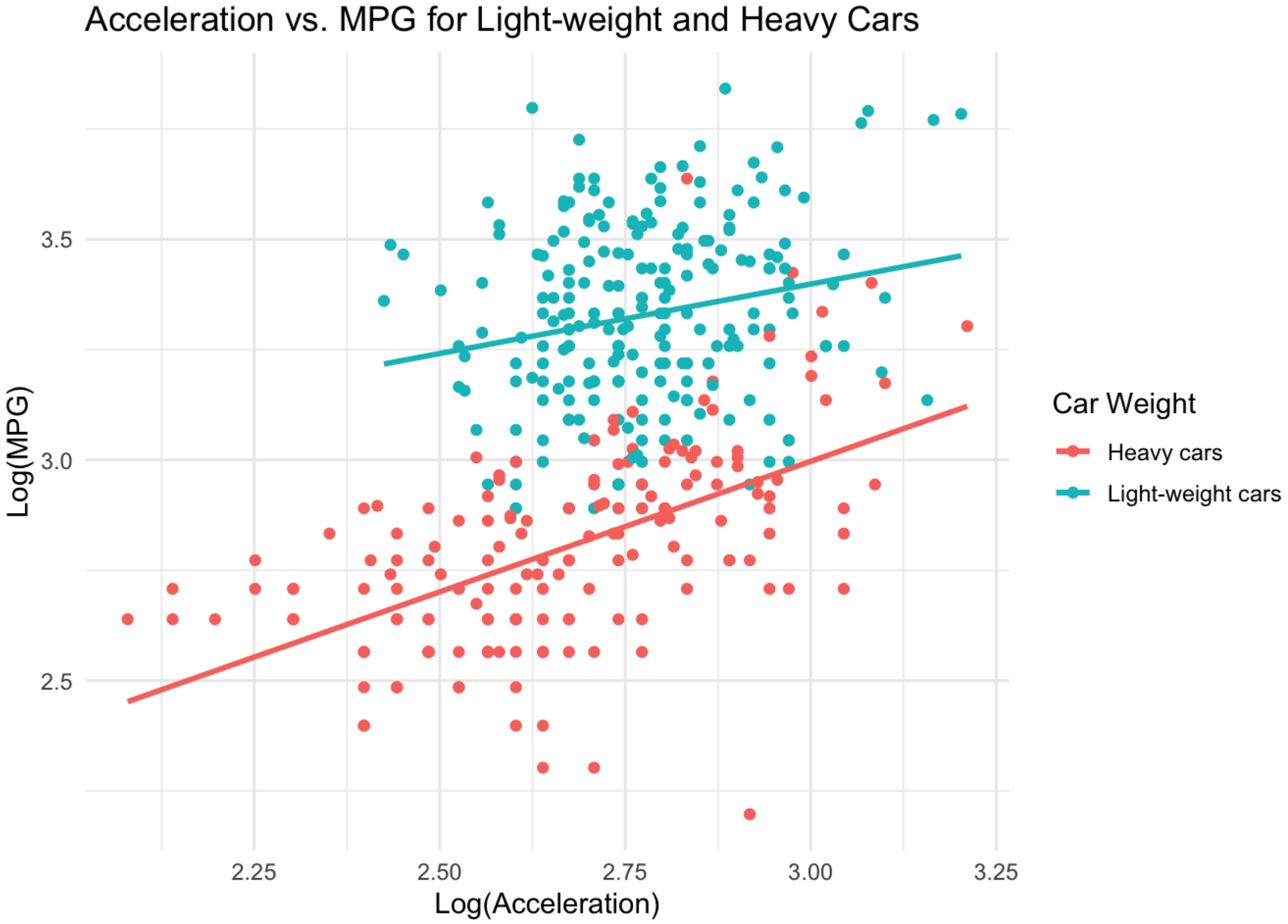


iii. Draw two slopes of acceleration-vs-mpg over the scatter plot: one slope for light cars and one slope for heavy cars (distinguish them by appearance)## R

Markdown

```
# Add a weight_category column to cars_log dataset
cars_log$weight_category <- ifelse(cars_log$weight < log(mean_weight), "Light-weight cars", "Heavy cars")

# Create a scatter plot of acceleration vs. mpg with different colors for light and heavy cars
plot <- ggplot(data = cars_log, aes(x = acceleration, y = mpg, color = weight_category)) +
  geom_point() +
  labs(title = "Acceleration vs. MPG for Light-weight and Heavy Cars",
       x = "Log(Acceleration)",
       y = "Log(MPG)",
       color = "Car Weight") +
  theme_minimal()

# Add two slopes for light-weight and heavy cars
plot_with_slopes <- plot +
  geom_smooth(data = subset(cars_log, weight_category == "Light-weight cars"),
              aes(x = acceleration, y = mpg, color = weight_category),
              method = "lm", se = FALSE, linetype = "solid") +
  geom_smooth(data = subset(cars_log, weight_category == "Heavy cars"),
              aes(x = acceleration, y = mpg, color = weight_category),
              method = "lm", se = FALSE, linetype = "solid")

# Display the plot
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

```
# Perform separate linear regressions for light and heavy cars
light_cars_regression <- lm(mpg ~ weight + acceleration + model_year + origin, data = light_cars)
heavy_cars_regression <- lm(mpg ~ weight + acceleration + model_year + origin, data = heavy_cars)

# Display full summaries of the linear regressions
summary(light_cars_regression)
```

```


Call:
lm(formula = mpg ~ weight + 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 ***
weight       -0.849942   0.056655 -15.002   <2e-16 ***
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(heavy_cars_regression)
```

```

Call:
lm(formula = mpg ~ weight + 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 ***
weight       -0.822352   0.077206 -10.651  < 2e-16 ***
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?

Based on the regression results, the following observations can be made about light versus heavy cars:

1. In both models, the `weight` variable has a significant negative effect on `mpg`. This indicates that as the weight of the cars increases, the fuel efficiency (mpg) decreases. The effect is stronger for light cars (-0.8499) compared to heavy cars (-0.8224), meaning that an increase in weight has a more significant impact on the mpg of lighter cars.
2. The `model_year` variable also has a significant positive effect on `mpg` in both models. This suggests that more recent car models tend to have better fuel efficiency (mpg). The effect is stronger for light cars (0.0329) compared to heavy cars (0.0303), indicating that newer models of lighter cars have improved more in fuel efficiency over time than heavier cars.

3. The `acceleration` variable has a positive effect on `mpg` , but it is only statistically significant for light cars (p-value = 0.0578). This means that faster acceleration tends to be associated with better fuel efficiency for light cars, while the relationship is not statistically significant for heavy cars.

4. The `origin` variable shows a significant positive effect on `mpg` for heavy cars (p-value = 0.0246) but not for light cars (p-value = 0.1698). This suggests that the origin of heavy cars has a more substantial impact on fuel efficiency compared to light cars.

Overall, the results indicate that both light and heavy cars share some similarities in the factors affecting their fuel efficiency (mpg). However, there are differences in the magnitude and statistical significance of these factors, suggesting that the dynamics of fuel efficiency might vary between light and heavy 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.

Based on the transformed dataset (cars_log) and considering weight and acceleration as potential factors affecting mileage (mpg), it is reasonable to hypothesize that:

1. Weight might be a moderating variable: The relationship between acceleration and mileage (mpg) could be different for light-weight and heavy cars. In other words, the effect of acceleration on mpg might be stronger or weaker depending on the weight of the car.

2. Acceleration would be an independent variable: Acceleration is directly related to how the car performs and is likely to have a direct impact on the car’s fuel efficiency (mpg) without necessarily depending on the weight of the car.

this is a hypothesis based on intuition and experience, and it should be tested statistically to validate or refute the assumptions.

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

```
model1 <- lm(mpg ~ weight + acceleration + model_year + origin, data = cars_log)
summary(model1)
```

```

Call:
lm(formula = mpg ~ weight + 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 ***
weight       -0.889384   0.028466 -31.243   <2e-16 ***
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

```
model2 <- lm(mpg ~ weight * acceleration + model_year + origin, data = cars_log)
summary(model2)
```

```

Call:
lm(formula = mpg ~ weight * 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    
weight              -0.179842   0.339101  -0.530   0.5962    
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 .  
weight: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 weight and acceleration
cars_log$centered_weight <- cars_log$weight - mean(cars_log$weight)
cars_log$centered_acceleration <- cars_log$acceleration - mean(cars_log$acceleration)

model3 <- lm(mpg ~ centered_weight * centered_acceleration + model_year + origin, data = cars_log)
summary(model3)
```

```

Call:
lm(formula = mpg ~ centered_weight * centered_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)                            0.566397   0.132258   4.283 2.33e-05 ***
centered_weight                       -0.893616   0.028415 -31.448  < 2e-16 ***
centered_acceleration                  0.082003   0.037725   2.174   0.0303 *  
model_year                             0.032933   0.001728  19.057  < 2e-16 ***
origin                                 0.016595   0.009164   1.811   0.0709 .  
centered_weight:centered_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
```

iv. Report a regression with an orthogonalized interaction term

```
# Orthogonalize weight and acceleration
cars_log$orth_weight <- cars_log$weight - cor(cars_log$weight, cars_log$acceleration) * cars_log$acceleration

model4 <- lm(mpg ~ orth_weight * acceleration + model_year + origin, data = cars_log)
summary(model4)
```

```

Call:
lm(formula = mpg ~ orth_weight * acceleration + model_year + 
    origin, data = cars_log)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.38895 -0.07057 -0.00121  0.06890  0.39583 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)               5.230718   3.706090   1.411   0.1589    
orth_weight              -0.639701   0.400397  -1.598   0.1109    
acceleration              1.282618   1.347167   0.952   0.3416    
model_year                0.032321   0.001726  18.728   <2e-16 ***
origin                    0.017707   0.009230   1.919   0.0558 .  
orth_weight:acceleration -0.091697   0.146675  -0.625   0.5322    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1165 on 392 degrees of freedom
Multiple R-squared:  0.8838,	Adjusted R-squared:  0.8823 
F-statistic: 596.1 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?

1. Raw interaction term (weight * acceleration):

```
cor(cars_log$weight, cars_log$weight * cars_log$acceleration)
```

```
[1] 0.1083055
```

```
cor(cars_log$acceleration, cars_log$weight * cars_log$acceleration)
```

```
[1] 0.852881
```

2. Mean-centered interaction term (centered_weight * centered_acceleration):

```
cor(cars_log$centered_weight, cars_log$centered_weight * cars_log$centered_acceleration)
```

```
[1] -0.2026948
```

```
cor(cars_log$centered_acceleration, cars_log$centered_weight * cars_log$centered_acceleration)
```

```
[1] 0.3512271
```

3. Orthogonalized interaction term (orth_weight * acceleration):

```
cor(cars_log$orth_weight, cars_log$orth_weight * cars_log$acceleration)
```

```
[1] 0.2529945
```

```
cor(cars_log$acceleration, cars_log$orth_weight * cars_log$acceleration)
```

```
[1] 0.9120915
```

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)

```
model1_direct <- lm(weight ~ cylinders, data = cars_log)
summary(model1_direct)
```

```
Call:
lm(formula = weight ~ cylinders, data = cars_log)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.35931 -0.09070 -0.00449  0.09822  0.34739 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 7.162982   0.022316  320.98   <2e-16 ***
cylinders   0.145544   0.003906   37.26   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1324 on 396 degrees of freedom
Multiple R-squared:  0.7781,	Adjusted R-squared:  0.7775 
F-statistic:  1388 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)

```
model2_direct <- lm(mpg ~ weight + acceleration + model_year + origin, data = cars_log)
summary(model2_direct)
```

```
Call:
lm(formula = mpg ~ weight + 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 ***
weight       -0.889384   0.028466 -31.243   <2e-16 ***
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
```

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

Model 1: log.weight ~ log.cylinders Coefficient (slope) for log.cylinders: 0.145544

Model 2: log.mpg ~ log.weight + log.acceleration + model_year + origin Coefficient (slope) for log.weight: -0.889384

Indirect effect = (slope of log.cylinders in Model 1) * (slope of log.weight in Model 2) = 0.145544 * (-0.889384) = -0.129479

The indirect effect of log.cylinders on log.mpg through log.weight is -0.129479.

This indicates that as the number of cylinders increases, the mpg decreases indirectly through the effect of increased weight.

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.?

```r
library(boot)
```

```


Attaching package: 'boot'
```

```
The following object is masked from 'package:psych':

    logit
```

```
The following object is masked from 'package:car':

    logit
```

```
# Define function to compute indirect effects
indirect_effect <- function(data, indices) {
  data <- data[indices,]

  model1 <- lm(weight ~ cylinders, data = data)
  model2 <- lm(mpg ~ weight + acceleration + model_year + origin, data = data)

  indirect <- coef(model1)["cylinders"] * coef(model2)["weight"]

  return(indirect)
}

set.seed(12345)  # You can choose any number as the seed
#Boot strapped for 10000 times
boot_results <- boot(data = cars_log, statistic = indirect_effect, R = 10000)

# Compute 95% CI
boot_ci <- boot.ci(boot.out = boot_results, conf = 0.95, type = "perc")
boot_ci
```

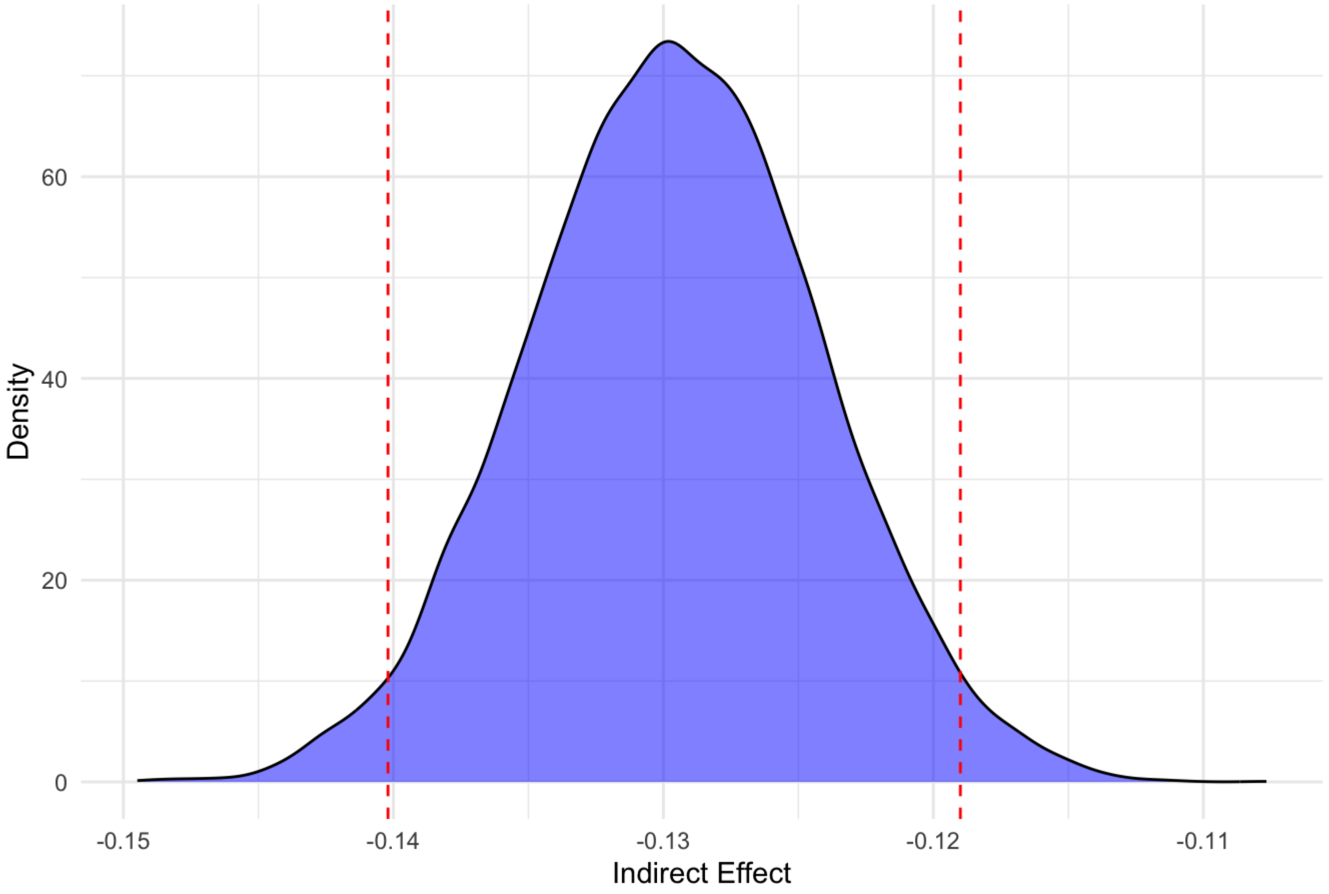
```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = boot_results, conf = 0.95, type = "perc")

Intervals :
Level     Percentile
95%   (-0.1402, -0.1190 )
Calculations and Intervals on Original Scale
```

The 95% confidence interval using the percentile method is given as (-0.1402, -0.1190). This means that we can be 95% confident that the true value of the statistic falls within this interval. Note that this is an estimate, and the true value may still fall outside this range.

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

```
#extracts the indirect effect
indirect_effects <- boot_results$t
# Create a density plot of the indirect effects
ggplot(data.frame(indirect_effects), aes(x=indirect_effects)) +
  geom_density(fill="blue", alpha=0.5) +
  geom_vline(aes(xintercept=-0.1402), color="red", linetype="dashed") +
  geom_vline(aes(xintercept=-0.1190), color="red", linetype="dashed") +
  labs(title="Density Plot of the 95% CI of the Indirect Effect",
       x="Indirect Effect",
       y="Density") +
  theme_minimal()
```

Density Plot of the 95% CI of the Indirect Effect


Density
Indirect Effect