HW10

113078506

2025-04-24

- Gen AI Usage: I use Gen AI to refine my grammar, verify my reasoning and adjust to cleaner code.
- Students who helped: 113078505, 113078502, and 113078514, helped with conclusion reasoning.

Set Up

```
raw_cars <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(raw_cars) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")

cars <- with(raw_cars, data.frame(mpg, weight, acceleration, model_year, origin))
cars_log <- with(cars, data.frame(log(mpg), log(weight), log(acceleration), model_year, origin))
```

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:

```
# Calculate mean of log.weight
mean_log_weight <- mean(cars_log$log.weight., na.rm = TRUE)

# Subset data
light_cars <- subset(cars_log, log.weight. < mean_log_weight)
heavy_cars <- subset(cars_log, log.weight. >= mean_log_weight)

cat("light cars:\n"); print(head(light_cars, 5))
```

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 how you might compare log weights to mean weight

```
## light cars:

##    log.mpg. log.weight. log.acceleration. model_year origin
## 15 3.178054    7.771489          2.708050         70      3
## 16 3.091042    7.949091          2.740840         70      1
## 17 2.890372    7.928046          2.740840         70      1
## 18 3.044522    7.858254          2.772589         70      1
## 19 3.295837    7.663877          2.674149         70      3
```

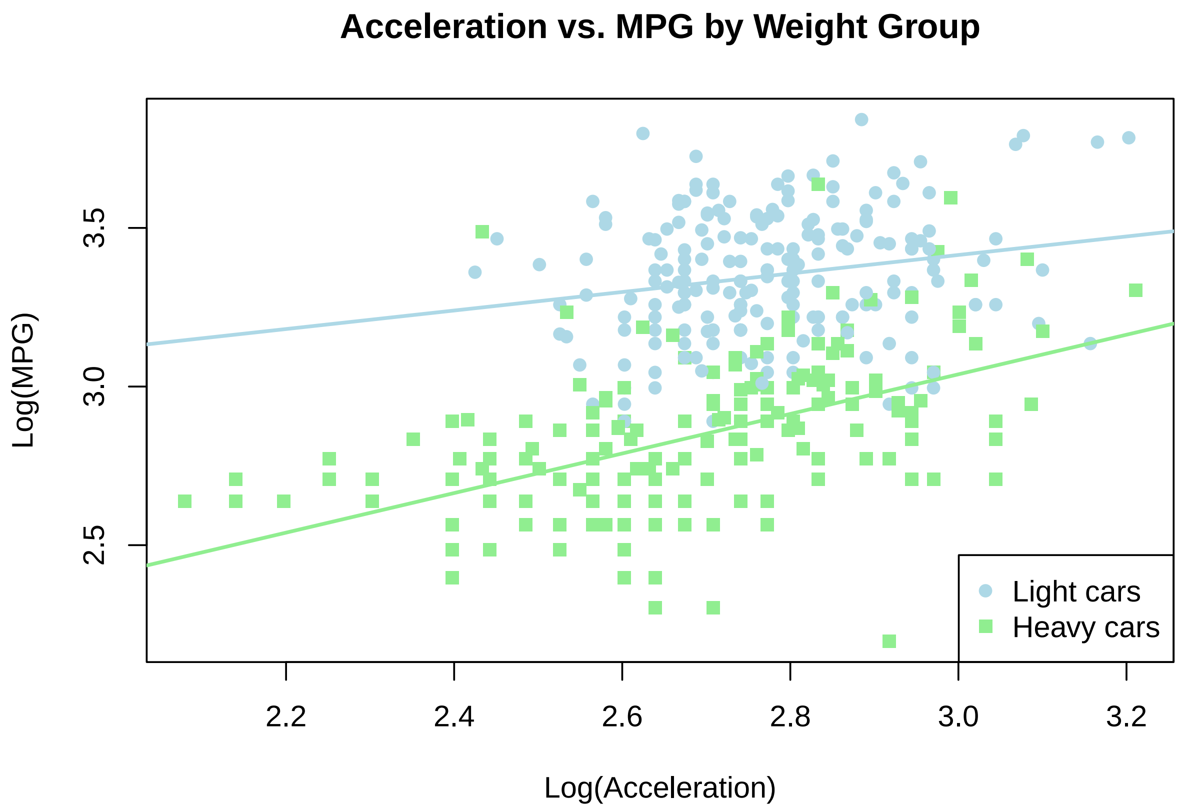
```r
cat("heavy cars:\n"); print(head(heavy_cars, 5))
```

```
## heavy cars:

##   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
```

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

```r
cars_log$weight_group <- ifelse(cars_log$log.weight. < mean_log_weight, "Light", "Heavy")

colors <- c("Light" = "lightblue", "Heavy" = "lightgreen")
shapes <- c("Light" = 16, "Heavy" = 15)

plot(cars_log$log.acceleration., cars_log$log.mpg.,
     col = colors[cars_log$weight_group],
     pch = shapes[cars_log$weight_group],
     xlab = "Log(Acceleration)",
     ylab = "Log(MPG)",
     main = "Acceleration vs. MPG by Weight Group")

legend("bottomright", legend = c("Light cars", "Heavy cars"),
       col = c("lightblue", "lightgreen"), pch = c(16, 15))

model_light <- lm(log.mpg. ~ log.acceleration., data = light_cars)
abline(model_light, col = colors["Light"], lwd = 2)
model_heavy <- lm(log.mpg. ~ log.acceleration., data = heavy_cars)
abline(model_heavy, col = colors["Heavy"], lwd = 2)
```

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)



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

```
# Regression for light cars
model_light_full <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year + factor(origin), data =

# Regression for heavy cars
model_heavy_full <- lm(log.mpg. ~ log.weight. + log.acceleration. + model_year + factor(origin), data =

summary(model_light_full)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = light_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36590 -0.06612  0.00637  0.06333  0.31513
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6.809014   0.598446  11.378   <2e-16 ***
```

```
## log.weight.       -0.821951   0.065769 -12.497   <2e-16 ***
## log.acceleration.  0.111137   0.058297   1.906   0.0580 .
## model_year         0.033344   0.002049  16.270   <2e-16 ***
## factor(origin)2    0.042309   0.020926   2.022   0.0445 *
## factor(origin)3    0.020923   0.019210   1.089   0.2774
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1102 on 199 degrees of freedom
## Multiple R-squared:  0.7093, Adjusted R-squared:  0.702
## F-statistic:  97.1 on 5 and 199 DF,  p-value: < 2.2e-16
```

```r
summary(model_heavy_full)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = heavy_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37099 -0.07224  0.00150  0.06704  0.42751
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.132892   0.677740  10.525  < 2e-16 ***
## log.weight.       -0.825517   0.068101 -12.122  < 2e-16 ***
## log.acceleration.  0.031221   0.055465   0.563  0.57418
## model_year         0.031735   0.003254   9.752  < 2e-16 ***
## factor(origin)2    0.099027   0.033840   2.926  0.00386 **
## factor(origin)3    0.063148   0.065535   0.964  0.33650
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1212 on 187 degrees of freedom
## Multiple R-squared:  0.7585, Adjusted R-squared:  0.752
## F-statistic: 117.4 on 5 and 187 DF,  p-value: < 2.2e-16
```

c. Using your intuition only: What do you observe about light versus heavy cars so far?

- Light cars generally have higher mpg than heavy cars at the same level of acceleration (their points are located higher on the y-axis). The intercept difference between the two groups is obvious, indicating that weight is an important factor that shifts the baseline fuel efficiency downward for heavier vehicles.

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.

- Intuitively, weight may act as a moderating variable that influences how strongly acceleration affects fuel efficiency (mpg), while acceleration itself serves as an independent variable directly related to mpg.

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)

```
summary(lm( log.mpg. ~ log.weight. + log.acceleration. + model_year + factor(origin), data = cars_log))
```

i. Report a regression without any interaction terms

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38275 -0.07032  0.00491  0.06470  0.39913
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.431155   0.312248  23.799  < 2e-16 ***
## log.weight.       -0.876608   0.028697 -30.547  < 2e-16 ***
## log.acceleration.  0.051508   0.036652   1.405  0.16072
## model_year         0.032734   0.001696  19.306  < 2e-16 ***
## factor(origin)2    0.057991   0.017885   3.242  0.00129 **
## factor(origin)3    0.032333   0.018279   1.769  0.07770 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1156 on 392 degrees of freedom
## Multiple R-squared:  0.8856, Adjusted R-squared:  0.8841
## F-statistic: 606.8 on 5 and 392 DF,  p-value: < 2.2e-16
```

```
cars_log$interaction_raw <- cars_log$log.weight. * cars_log$log.acceleration.
summary(lm(log.mpg. ~ log.weight. + log.acceleration. + interaction_raw + model_year + factor(origin), d
```

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

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + interaction_raw +
##     model_year + factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37807 -0.06868  0.00463  0.06891  0.39857
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.089642   2.752872   0.396  0.69245
## log.weight.        -0.096632   0.337637  -0.286  0.77488
## log.acceleration.   2.357574   0.995349   2.369  0.01834 *
## interaction_raw    -0.287170   0.123866  -2.318  0.02094 *
## model_year          0.033685   0.001735  19.411  < 2e-16 ***
## factor(origin)2     0.058737   0.017789   3.302  0.00105 **
## factor(origin)3     0.028179   0.018266   1.543  0.12370
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.115 on 391 degrees of freedom
## Multiple R-squared:  0.8871, Adjusted R-squared:  0.8854
## F-statistic: 512.2 on 6 and 391 DF,  p-value: < 2.2e-16
```

```r
accel_mc <- scale(cars_log$log.acceleration., center=TRUE, scale=FALSE)
weight_mc <- scale(cars_log$log.weight., center=TRUE, scale=FALSE)
cars_log$meancenter <- accel_mc*weight_mc

summary(lm(cars_log$log.acceleration. ~ accel_mc + weight_mc + accel_mc*weight_mc))
```

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

```
##
## Call:
## lm(formula = cars_log$log.acceleration. ~ accel_mc + weight_mc +
##     accel_mc * weight_mc)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.478e-14  1.100e-17  1.520e-16  2.960e-16  1.063e-15
##
## Coefficients:
##                      Estimate Std. Error    t value Pr(>|t|)
## (Intercept)         2.729e+00  1.800e-16  1.516e+16   <2e-16 ***
## accel_mc            1.000e+00  1.054e-15  9.488e+14   <2e-16 ***
## weight_mc          -1.162e-16  6.489e-16 -1.790e-01    0.858
## accel_mc:weight_mc  3.147e-16  3.416e-15  9.200e-02    0.927
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.277e-15 on 394 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 4.024e+29 on 3 and 394 DF,  p-value: < 2.2e-16
```

```r
accel_x_weight <- cars_log$log.acceleration. * cars_log$log.weight.
interaction_regr <- lm(accel_x_weight ~ cars_log$log.acceleration. + cars_log$log.weight.)
interaction_ortho <- interaction_regr$residuals
```

```r
summary(lm(log.mpg. ~ log.acceleration. + log.weight. + interaction_ortho, data=cars_log))
```

iv. Report a regression with an orthogonalized interaction term

```
##
## Call:
## lm(formula = log.mpg. ~ log.acceleration. + log.weight. + interaction_ortho,
##     data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49728 -0.10145 -0.01102  0.09665  0.56416
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.48669    0.33430  31.369  < 2e-16 ***
## log.acceleration.  0.21084    0.04949   4.260 2.56e-05 ***
## log.weight.       -1.00048    0.03187 -31.395  < 2e-16 ***
## interaction_ortho  0.25295    0.16807   1.505    0.133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1613 on 394 degrees of freedom
## Multiple R-squared:  0.7763, Adjusted R-squared:  0.7746
## F-statistic: 455.7 on 3 and 394 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?

```r
library(knitr)
# Correlation between raw interaction term and its components
raw_interact_data <- data.frame(
  log_weight = cars_log$log.weight.,
  log_acceleration = cars_log$log.acceleration.,
  interaction = cars_log$interaction_raw
)

# Now calculate the correlation matrix
cor_matrix <- cor(raw_interact_data)

knitr::kable(cor_matrix, caption = "Correlations between raw interaction and IV")
```

Table 1: Correlations between raw interaction and IV

	log_weight	log_acceleration	interaction
log_weight	1.0000000	-0.4256194	0.1083055
log_acceleration	-0.4256194	1.0000000	0.8528810
interaction	0.1083055	0.8528810	1.0000000

```r
# Correlation between mean-centered interaction term and its components
mc_interact_data <- data.frame(
  log_weight = cars_log$log.weight.,
  log_acceleration = cars_log$log.acceleration.,
  mc_interaction = cars_log$meancenter
)

# Now calculate the correlation matrix
cor_matrix_mc <- cor(mc_interact_data)

# Display the correlation matrix

knitr::kable(cor_matrix_mc, caption = "Correlations between mean-centered interaction and IV")
```

Table 2: Correlations between mean-centered interaction and IV

	log_weight	log_acceleration	mc_interaction
log_weight	1.0000000	-0.4256194	-0.2026948
log_acceleration	-0.4256194	1.0000000	0.3512271
mc_interaction	-0.2026948	0.3512271	1.0000000

```r
# Correlation between orthogonalized interaction term and its components
ortho_interact_data <- data.frame(
  log_weight = cars_log$log.weight.,
  log_acceleration = cars_log$log.acceleration.,
  ortho_interaction = interaction_ortho
)

# Now calculate the correlation matrix
cor_matrix_ortho <- cor(ortho_interact_data)

# Display the correlation matrix

knitr::kable(cor_matrix_ortho, caption = "Correlations between orthogonalized interaction and IV")
```

Table 3: Correlations between orthogonalized interaction and IV

	log_weight	log_acceleration	ortho_interaction
log_weight	1.0000000	-0.4256194	0
log_acceleration	-0.4256194	1.0000000	0
ortho_interaction	0.0000000	0.0000000	1

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)

```
cars_log$log.cylinders. <- log(raw_cars$cylinders)

ml_1 <- lm(log.weight.~log.cylinders., data=cars_log)
summary(ml_1)
```

```
##
## 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)

```
ml_2 <- lm(log.mpg.~log.weight. + log.acceleration. + model_year + origin, data=cars_log)
summary(ml_2)
```

```
##
## 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
```

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

```
slope_ml1 <- 0.82012
slope_ml2 <- -0.889384

indirect_effect <- slope_ml1 * slope_ml2
print(indirect_effect)
```

```
## [1] -0.7294016
```

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.?

```
boot_mediation <- function(model1, model2, dataset) {
boot_index <- sample(1:nrow(dataset), replace=TRUE)
data_boot <- dataset[boot_index, ]
regr1 <- lm(model1, data_boot)
regr2 <- lm(model2, data_boot)
return(regr1$coefficients[2] * regr2$coefficients[2]) # indirect effect
}

set.seed(42)
indirect <- replicate(2000, boot_mediation(ml_1, ml_2, cars_log))
quantile(indirect, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.7893935 -0.6719537
```

```
ci <- quantile(indirect, probs=c(0.025, 0.975))
plot(density(indirect), main = "Bootstrap Distribution of Indirect Effect", xlab = "Indirect Effect")
abline(v = ci, col = "blue", lty = "dashed")
```

ii. Show a density plot of the distribution of the indirect effect, and mark its 95% CI