HW12

110078509

20220508

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)

Question 1

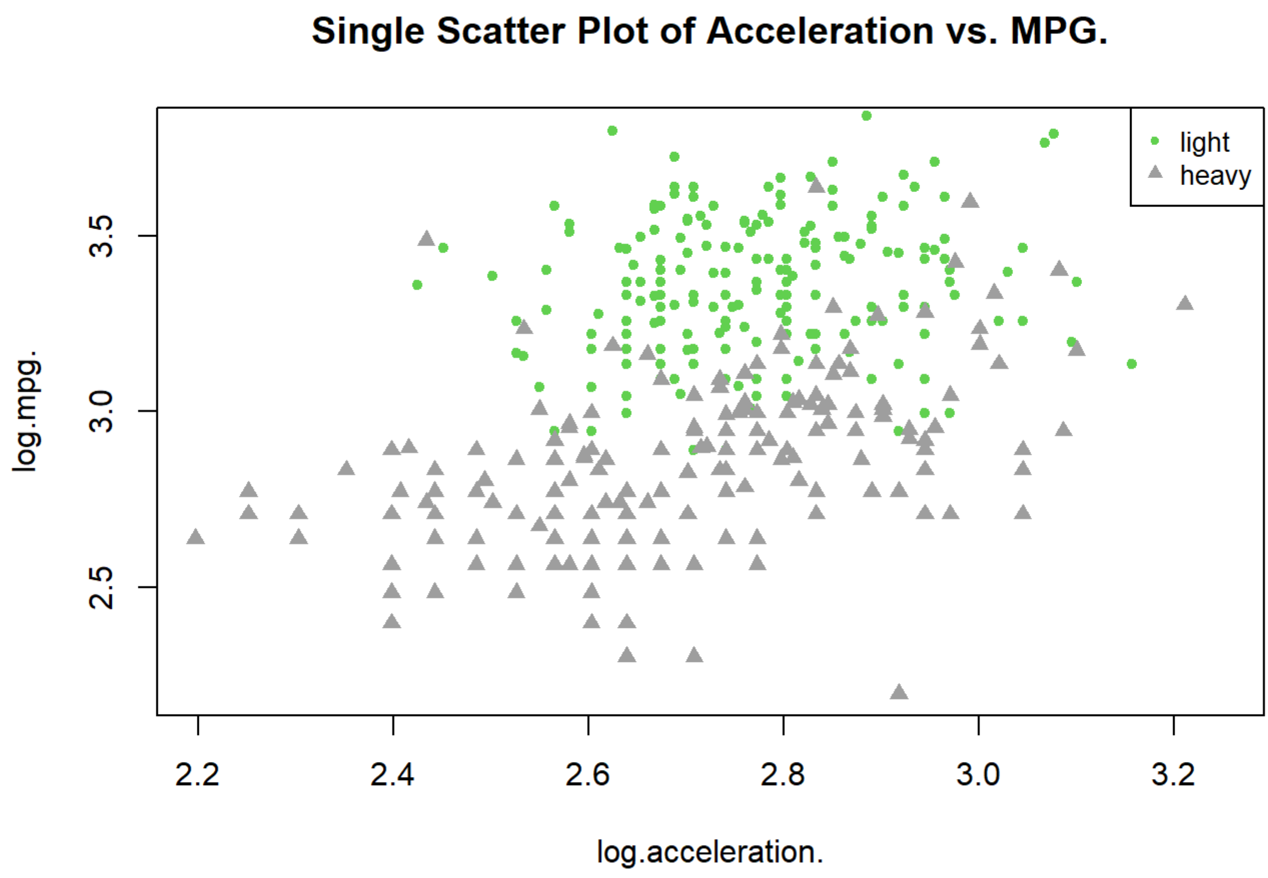
a. Let's visualize how weight might moderate the relationship between acceleration and mpg:

ai. 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

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

cars_log.2 <- with(cars, data.frame(log(mpg),log(cylinders),log(displacement),log(horsepowe
r), log(weight), log(acceleration), model_year, origin))

cars_log <- with(cars, data.frame(log(mpg), log(weight), log(acceleration), model_year, origi
n))


carmean<-mean(cars_log$log.weight.)

light <- subset(cars_log, log.weight.<carmean, na.action=na.exclude)
heavy <- subset(cars_log, log.weight.>carmean, na.action=na.exclude)
```

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

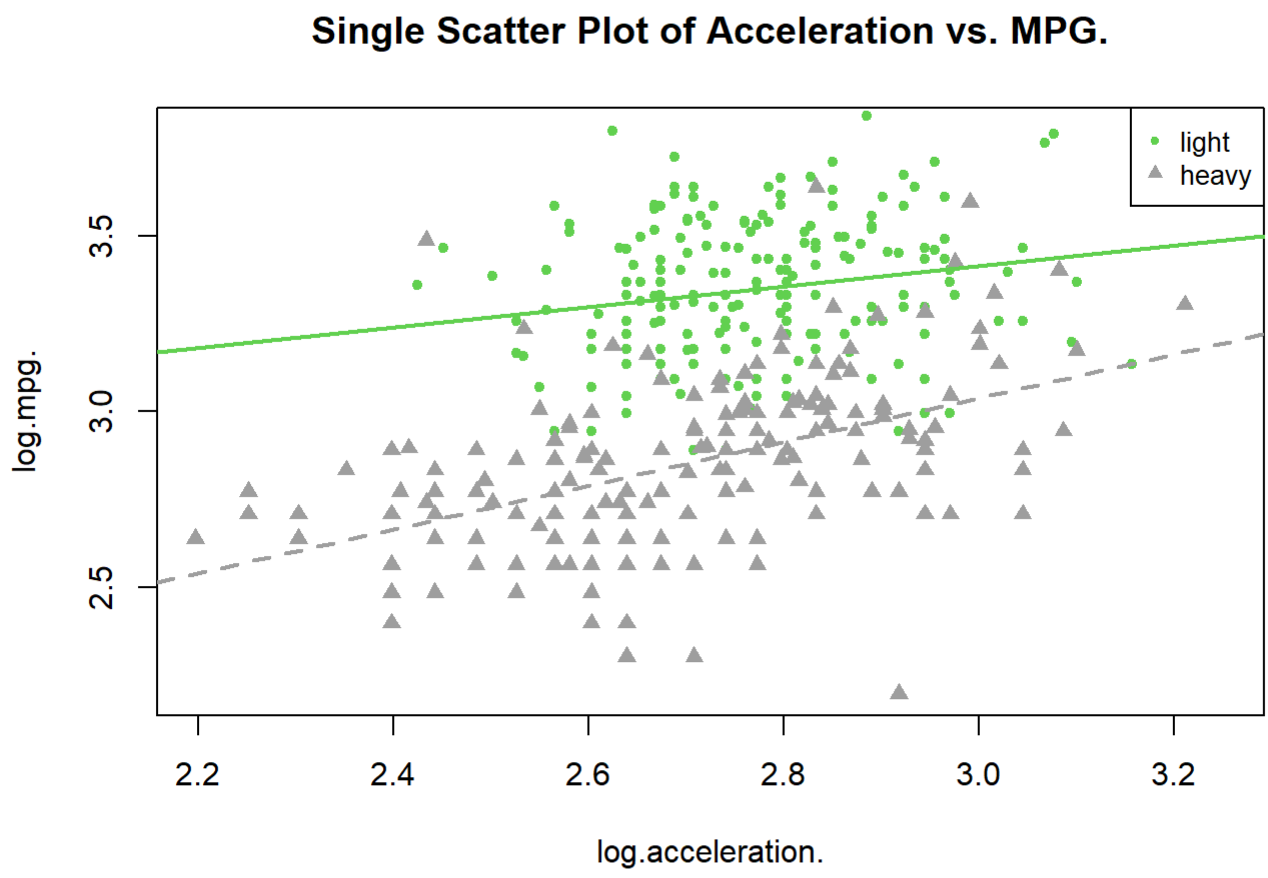
```
with(light, plot(log.acceleration.,xlim=c(2.2,3.25) ,ylim = c(2.2,3.8), log.mpg., pch=20, col
=3, main='Single Scatter Plot of Acceleration vs. MPG.'))

with(heavy, points(log.acceleration., log.mpg., pch=17, col=8))

legend('topright', legend=c("light", "heavy"),
        pch=c(20,17), cex=0.85, col=c(3,8))
```



aiii. 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)

```
with(light, plot(log.acceleration., log.mpg.,xlim=c(2.2,3.25),ylim=c(2.2,3.8), pch=20, col=3,
main='Single Scatter Plot of Acceleration vs. MPG.'))

with(heavy, points(log.acceleration., log.mpg., pch=17, col=8))
abline(lm(log.mpg. ~ log.acceleration., data=light), col=3, lwd=2)
abline(lm(log.mpg. ~ log.acceleration., data=heavy), col=8, lwd=2, lty=2)

legend('topright', legend=c("light", "heavy"), pch=c(20,17), cex=0.85,col=c(3,8))
```



b. Report the full summaries of two separate regressions for light and heavy cars where

- Ans:

```
light<-na.omit(light)
heavy<-na.omit(heavy)

print('Light:')
```

```
## [1] "Light:"
```

```
light.lm <- lm(light$log.mpg.~light$log.weight.+light$log.acceleration.+light$model_year+fact
or(light$origin))
summary(light.lm)
```

```
##
## Call:
## lm(formula = light$log.mpg. ~ light$log.weight. + light$log.acceleration. +
##     light$model_year + factor(light$origin))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36590 -0.06612  0.00637  0.06333  0.31513
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.809014   0.598446  11.378   <2e-16 ***
## light$log.weight.       -0.821951   0.065769 -12.497   <2e-16 ***
## light$log.acceleration.  0.111137   0.058297   1.906   0.0580 .
## light$model_year         0.033344   0.002049  16.270   <2e-16 ***
## factor(light$origin)2    0.042309   0.020926   2.022   0.0445 *
## factor(light$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
```

```
print('----------------------------BLOCK----------------------------------------')
```

```
## [1] "----------------------------BLOCK----------------------------------------"
```

```
print('Heavy:')
```

```
## [1] "Heavy:"
```

```
heavy.lm <- lm(heavy$log.mpg.~heavy$log.weight.+heavy$log.acceleration.+heavy$model_year+fact
or(heavy$origin))
summary(heavy.lm)
```

```
##
## Call:
## lm(formula = heavy$log.mpg. ~ heavy$log.weight. + heavy$log.acceleration. +
##     heavy$model_year + factor(heavy$origin))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37099 -0.07224  0.00150  0.06704  0.42751
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.132892   0.677740  10.525  < 2e-16 ***
## heavy$log.weight.       -0.825517   0.068101 -12.122  < 2e-16 ***
## heavy$log.acceleration.  0.031221   0.055465   0.563  0.57418
## heavy$model_year         0.031735   0.003254   9.752  < 2e-16 ***
## factor(heavy$origin)2    0.099027   0.033840   2.926  0.00386 **
## factor(heavy$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. (not graded) Using your intuition only: What do you observe about light versus heavy cars so far?

- Ans:

log.acceleration. is only significant(at 0.1 significant.) in 'Light'.

Question 2

a. (not graded) Between weight and acceleration ability, use your intuition and experience to state which variable might be a moderating versus independent variable, in affecting mileage.

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

```
##
## 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
```

Personally, I guessed “log.weight.” could be the moderator because it’s slope is steeper than the “log.acceleration.”’s.

For example, assuming we’re testing the performance of the student adopt different learning platforms. In the example case like this , the “original academic performance” could be a strong moderator. Because the students who have a better original academic performance are more likely to have a strong learning ability. And in this case, the relationship between the “original academic performance (moderator)” & “performance after adopt learning platform (Y)” could be highly correlative, which the slope could be steeper.

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)

bi. Report a regression without any interaction terms

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

```
##
## 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
```

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

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

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + log.weight. *
##     log.acceleration. + 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 *
## 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
## log.weight.:log.acceleration. -0.287170   0.123866  -2.318  0.02094 *
## ---
## 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
```

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

```
slw <- scale(cars_log$log.weight., center=TRUE, scale=FALSE)
sla <- scale(cars_log$log.acceleration., center=TRUE, scale=FALSE)


summary(lm(log.mpg. ~ slw + sla +model_year + factor(origin)+ slw*sla, data=cars_log ))
```

```
##
## Call:
## lm(formula = log.mpg. ~ slw + sla + model_year + factor(origin) +
##     slw * sla, 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)       0.518882   0.132944   3.903 0.000112 ***
## slw              -0.880393   0.028585 -30.799  < 2e-16 ***
## sla               0.072596   0.037567   1.932 0.054031 .
## model_year        0.033685   0.001735  19.411  < 2e-16 ***
## factor(origin)2   0.058737   0.017789   3.302 0.001049 **
## factor(origin)3   0.028179   0.018266   1.543 0.123704
## slw:sla          -0.287170   0.123866  -2.318 0.020943 *
## ---
## 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
```

biv. Report a regression with an orthogonalized interaction term

```
# Residuals of interaction’s regression

log.weight_x_log.acceleration. <- cars_log$log.weight.*cars_log$log.acceleration.
interaction_regr <- lm(log.weight_x_log.acceleration. ~ cars_log$log.weight.+cars_log$log.acc
eleration.)

interaction_ortho <- interaction_regr$residuals
#Correlation of residual
#round(cor(cbind(dep, interaction_ortho)), 2)

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

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin) + interaction_ortho, 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)        7.377176   0.311392  23.691  < 2e-16 ***
## log.weight.       -0.876967   0.028539 -30.729  < 2e-16 ***
## log.acceleration.  0.046100   0.036524   1.262  0.20764
## 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
## interaction_ortho -0.287170   0.123866  -2.318  0.02094 *
## ---
## 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
```

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?

- Raw

```
# raw
weight_acce_raw <- cars_log$log.weight. * cars_log$log.acceleration.


round(cor(cbind(cars_log[1:3],weight_acce_raw)), 2)
```

```
##                   log.mpg. log.weight. log.acceleration. weight_acce_raw
## log.mpg.              1.00       -0.87              0.46            0.01
## log.weight.          -0.87        1.00             -0.43            0.11
## log.acceleration.     0.46       -0.43              1.00            0.85
## weight_acce_raw       0.01        0.11              0.85            1.00
```

- Mean-centered

```
# mean-centered
mean.center = cbind(cars_log$log.mpg., slw, sla, slw*sla)
colnames(mean.center) = c('log.mpg','scale_log.weight', 'scale_log.acceleration','interaction
_term' )

round(cor(mean.center), 2)
```

```
##                        log.mpg scale_log.weight scale_log.acceleration
## log.mpg                   1.00            -0.87                   0.46
## scale_log.weight         -0.87             1.00                  -0.43
## scale_log.acceleration    0.46            -0.43                   1.00
## interaction_term          0.24            -0.20                   0.35
##                        interaction_term
## log.mpg                            0.24
## scale_log.weight                  -0.20
## scale_log.acceleration             0.35
## interaction_term                   1.00
```

- Orthogonalized

```
# orthogonalized
round(cor(cbind(cars_log[1:3], interaction_ortho)), 2)
```

```
##                   log.mpg. log.weight. log.acceleration. interaction_ortho
## log.mpg.              1.00       -0.87              0.46              0.04
## log.weight.          -0.87        1.00             -0.43              0.00
## log.acceleration.     0.46       -0.43              1.00              0.00
## interaction_ortho     0.04        0.00              0.00              1.00
```

Question 3- Mediator

a.i. Model 1: Regress log.weight. over log.cylinders. only

(check whether number of cylinders has a significant direct effect on weight)

- Ans:

Yes, the log.cylinders. has a significant direct effect on log.weight. (p value < 0.05)

```
cars_log.2 <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement), log(horsepow
er), log(weight), log(acceleration), model_year,factor(origin)))
cars_log.2<-na.omit(cars_log.2)

model1 <-lm(log.weight.~log.cylinders., data= cars_log.2)
summary(model1)
```

```
##
## Call:
## lm(formula = log.weight. ~ log.cylinders., data = cars_log.2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35409 -0.09030 -0.00169  0.09271  0.40488
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.60059    0.03710  177.92   <2e-16 ***
## log.cylinders.  0.82187    0.02208   37.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1319 on 390 degrees of freedom
## Multiple R-squared:  0.7804, Adjusted R-squared:  0.7798
## F-statistic:  1386 on 1 and 390 DF,  p-value: < 2.2e-16
```

aii. 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?)

- Ans:

```
model2 <- lm(log.mpg. ~ log.weight.+log.acceleration.+model_year + factor(factor.origin.) , d
ata = cars_log.2)
summary(model2)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(factor.origin.), data = cars_log.2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38259 -0.07054  0.00401  0.06696  0.39798
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.410974   0.316806  23.393  < 2e-16 ***
## log.weight.             -0.875499   0.029086 -30.101  < 2e-16 ***
## log.acceleration.        0.054377   0.037132   1.464  0.14389
## model_year               0.032787   0.001731  18.937  < 2e-16 ***
## factor(factor.origin.)2  0.056111   0.018241   3.076  0.00225 **
## factor(factor.origin.)3  0.031937   0.018506   1.726  0.08519 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1163 on 386 degrees of freedom
## Multiple R-squared:  0.8845, Adjusted R-squared:  0.883
## F-statistic: 591.1 on 5 and 386 DF,  p-value: < 2.2e-16
```

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

```
indirect_effect_mpg.cylinder <- model1$coefficients[2]*model2$coefficients[2]
sprintf("The indirect Effect approximately equal to  %.2f", indirect_effect_mpg.cylinder)
```

```
## [1] "The indirect Effect approximately equal to  -0.72"
```

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

bi. 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.?

```
set.seed(74)
indirect<-replicate(2000,boot_mediation(model1,model2,cars_log.2))
quantile(indirect, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.7835367 -0.6582757
```

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

```
plot(density(indirect),lwd=2,col="blue", main='Bootstrapping of Indirect Effect')
abline(v=mean(indirect), lty=2, col="red", lwd=2)
abline(v=quantile(indirect, probs=c(0.025, 0.975)), lty=2, lwd=2, col="green")
```