HW12

107070008

```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement",
  "horsepower", "weight", "acceleration", "model_year", "origin", "car_name")
cars <- auto
cars <- cars[,-9]
cars <- cars[,-4]
cars <- cars[,-3]

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

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)

```
log_mean <- mean(cars_log$log.weight.)
light_cars <- subset(cars_log, log.weight. < log_mean)
head(light_cars)
```

```
##    log.mpg. log.weight. log.acceleration. model_year factor.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
## 20 3.258097    7.514800          3.020425         70              2
```

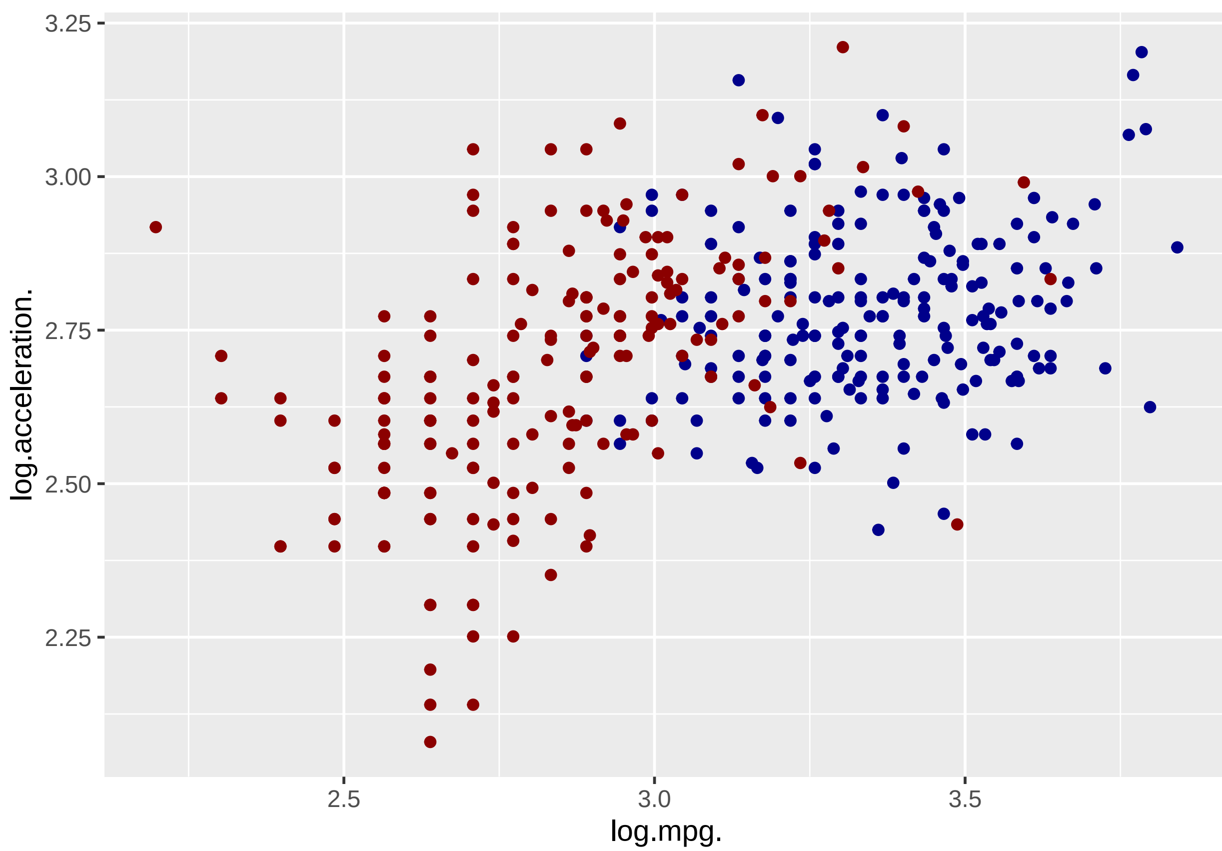
```
heavy_cars <- subset(cars_log, log.weight. >= log_mean)
head(heavy_cars)
```

```
##   log.mpg. log.weight. log.acceleration. model_year factor.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
```

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

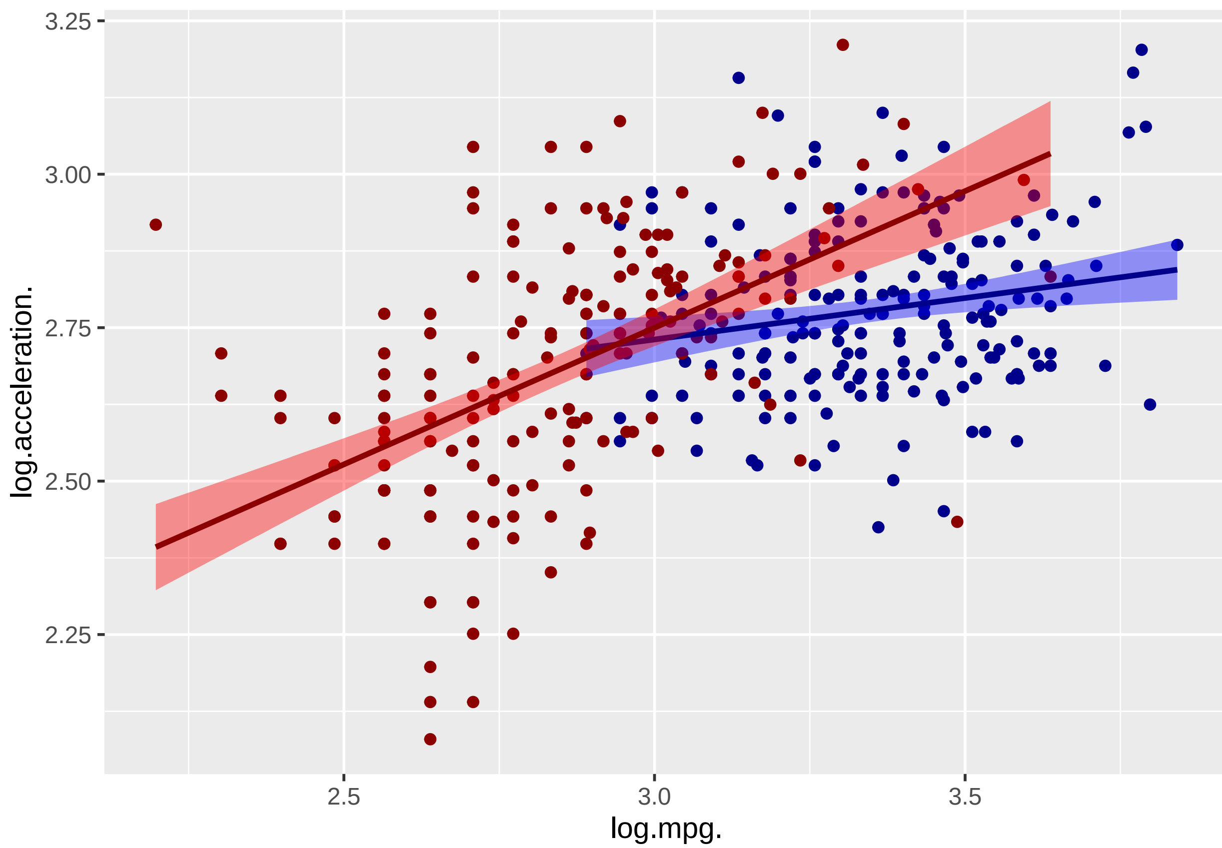
```
library(ggplot2)
p <- ggplot() +
  #light_cars
  geom_point(data = light_cars, aes(x = log.mpg., y = log.acceleration.),
             color='darkblue') +
  #heavy_cars
  geom_point(data = heavy_cars, aes(x = log.mpg., y = log.acceleration.),
              color='darkred')
p
```



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)

```
p + geom_smooth(data = light_cars, aes(x = log.mpg., y = log.acceleration.),
               fill="blue", colour="darkblue", method = "lm") +
    geom_smooth(data = heavy_cars, aes(x = log.mpg., y = log.acceleration.),
               fill="red", colour="darkred", method = "lm")
```

```
## `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

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

```
##
## 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
```

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

```
##
## 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 (not graded) Using your intuition only: What do you observe about light versus heavy cars so far?
`Their coefficient of acceleration change a lots.`

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

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

`I think that weight will become a moderate 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

```
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
```

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

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

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor.origin. + log.weight. * log.acceleration., 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
```

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

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

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

```
##
## Call:
## lm(formula = log.mpg. ~ weight_mc + accerlation_mc + model_year +
##     factor.origin. + weight_mc * accerlation_mc, 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 ***
## weight_mc                -0.880393   0.028585 -30.799  < 2e-16 ***
## accerlation_mc            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
## weight_mc:accerlation_mc -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
```

iv.Report a regression with an orthogonalized interaction term

```
w_X_a <- cars_log$log.weight.*cars_log$log.acceleration.
intereaction_regr <- lm(w_X_a ~ log.weight. + log.acceleration. + model_year +
                          factor.origin., data = cars_log)
interaction_ortho <- intereaction_regr$residuals
summary(lm(log.mpg. ~ log.weight. + log.acceleration. + interaction_ortho + model_year +
             factor.origin., data=cars_log))
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + interaction_ortho +
##     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)        7.431155   0.310520  23.931  < 2e-16 ***
## log.weight.       -0.876608   0.028538 -30.717  < 2e-16 ***
## log.acceleration.  0.051508   0.036450   1.413  0.15841
## interaction_ortho -0.287170   0.123866  -2.318  0.02094 *
## model_year         0.032734   0.001686  19.413  < 2e-16 ***
## factor.origin.2    0.057991   0.017786   3.260  0.00121 **
## factor.origin.3    0.032333   0.018178   1.779  0.07607 .
## ---
## 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?

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

```
## [1] 0.1083055
```

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

```
## [1] 0.852881
```

```
cor(weight_mc, weight_mc*accerlation_mc)
```

```
##            [,1]
## [1,] -0.2026948
```

```
cor(accerlation_mc, weight_mc*accerlation_mc)
```

```
##           [,1]
## [1,] 0.3512271
```

```
cor(cars_log$log.weight., interaction_ortho)
```

```
## [1] 2.084909e-17
```

```
cor(cars_log$log.acceleration., interaction_ortho)
```

```
## [1] 2.38378e-16
```

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?

```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement",
"horsepower", "weight", "acceleration", "model_year", "origin", "car_name")
cars <- auto
cars <- cars[,-9]
cars <- cars[,-4]
cars_log_ <- with(cars, data.frame(log(mpg), log(cylinders), log(weight), log(acceleration),
                  model_year, factor(origin)))
```

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)

```
summary(lm(log.weight. ~ log.cylinders., data = cars_log_))
```

```
##
## 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
```

Yes, it has 0.1% significant effect on weight, and its coefficient is 0.82012.

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?)

```r
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
```

Yes, it has 0.1% significant effect on weight, and its coefficient is -0.83628.

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

```r
wc_regr <- summary(lm(log.weight. ~ log.cylinders., data = cars_log_))
mw_regr <- summary(lm(log.mpg. ~ log.weight. + log.acceleration.
                  + model_year + factor.origin., data = cars_log_))
wc_regr$coefficients[2]*mw_regr$coefficients[2]
```

```
## [1] -0.7189275
```

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])
}
set.seed(42)
indirect <- replicate(2000, boot_mediation(wc_regr, mw_regr, cars_log_))
quantile(indirect, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.7784044 -0.6610106
```

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

```
plot(density(indirect), col = "blue")
abline(v=quantile(indirect, probs=c(0.025, 0.975)), col = "pink", lwd = 3)
```