hw12

111078513

Help By 108078467

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

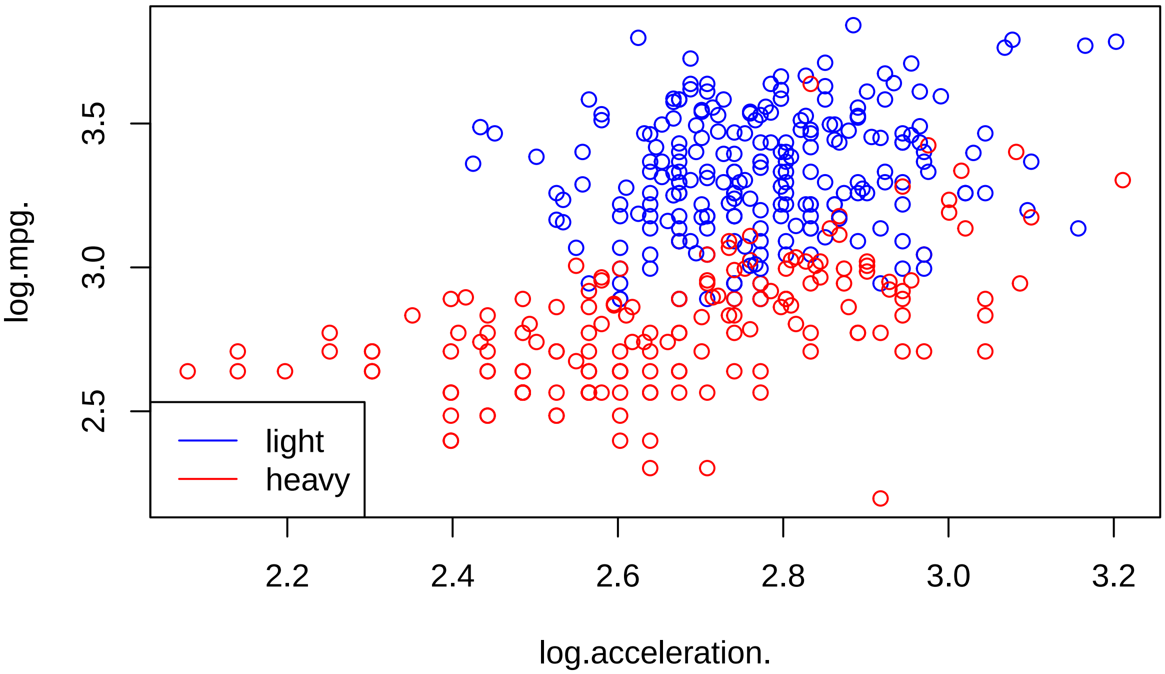
a. Let's visualize how weight might moderate the relationship between acceleration and mpg:

(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)

```
cars_log$weight_mask <- ifelse(cars$weight < mean(cars$weight), 1, 2)
cars_log_light <- cars_log[cars_log$weight_mask == 1, ]
cars_log_heavy <- cars_log[cars_log$weight_mask == 2, ]
```

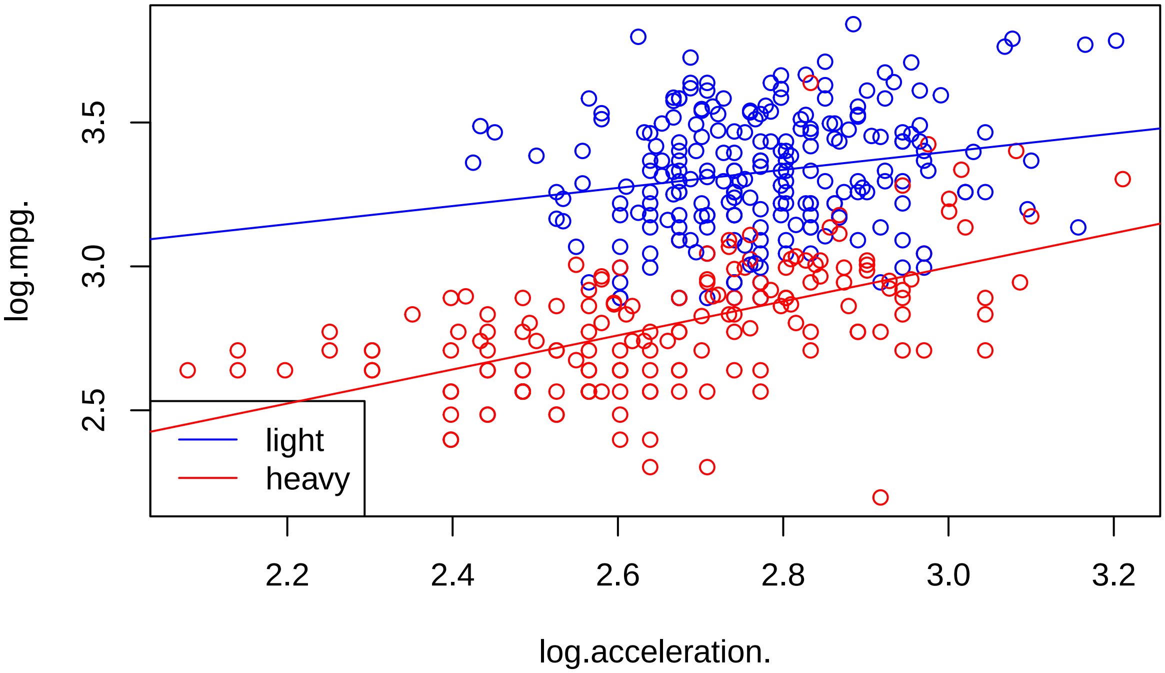
(ii) Create a single scatter plot of acceleration vs. mpg, with different colors and/or shapes for light versus heavy cars

```
weight_color = c("blue", "red")
with(cars_log, plot(log.acceleration., log.mpg.,  col=weight_color[weight_mask]))
legend("bottomleft", lty=1, c("light", "heavy"), col=weight_color)
```



(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)

```
cars_log_light_rg <- lm( log.mpg.~ log.acceleration., data=cars_log_light)
cars_log_heavy_rg <- lm(log.mpg.~ log.acceleration., data=cars_log_heavy)
with(cars_log, plot(log.acceleration., log.mpg.,  col=weight_color[weight_mask]))
legend("bottomleft", lty=1, c("light", "heavy"), col=weight_color)
abline(cars_log_light_rg, col = "blue")
abline(cars_log_heavy_rg, col = "red")
```



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

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

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     origin, data = cars_log_light)
##
## 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
```

```
cars_log_heavy_rg_full <- lm( log.mpg.~ log.weight.+ log.acceleration.+ model_year + origin
                              , data=cars_log_heavy)
summary(cars_log_light_rg_full)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     origin, data = cars_log_light)
##
## 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
```

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

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

```
cars_log$log.weight_mc <- scale(cars_log$log.weight., center = T, scale = F)
cars_log$log.acceleration_mc <- scale(cars_log$log.acceleration., center = T, scale = F)
summary(lm(log.mpg.~ log.weight_mc + log.acceleration_mc + model_year + factor(origin)
           +log.weight_mc * log.acceleration_mc, data=cars_log))
```

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

```
weight_x_acceleration <- cars_log$log.weight. * cars_log$log.acceleration.
interaction_regr <- lm(weight_x_acceleration ~ cars_log$log.weight.
                        + cars_log$log.acceleration.)
cars_log$interaction_ortho <- interaction_regr$residuals
summary(lm(log.mpg.~log.weight.+log.acceleration.+model_year+factor(origin)
           +interaction_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?

```
correlatin_df <- data.frame()
correlatin_df[1,1] <- with(cars_log, cor(log.weight., log.weight.*log.acceleration.))
correlatin_df[2,1] <- with(cars_log, cor(log.acceleration., log.weight.*log.acceleration.))
correlatin_df[1,2] <- with(cars_log, cor(log.weight_mc, log.weight_mc*log.acceleration_mc))
correlatin_df[2,2] <- with(cars_log, cor(log.acceleration_mc, log.weight_mc*log.acceleration_mc))
correlatin_df[1,3] <- with(cars_log, cor(log.weight.,interaction_ortho))
correlatin_df[2,3] <- with(cars_log, cor(log.acceleration.,interaction_ortho))
row.names(correlatin_df) <- c("weight","acceleration")
colnames(correlatin_df) <- c("raw","mean-centered","orthogonalized")
print(correlatin_df )
```

```
##                    raw mean-centered orthogonalized
## weight       0.1083055    -0.2026948   2.468461e-17
## acceleration 0.8528810     0.3512271  -6.804111e-17
```

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?

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

(i) Model 1: Regress log.weight. over log.cylinders. only

```
cars_log2 <- with(cars, data.frame(log(mpg), log(weight),log(cylinders), log(acceleration)
                                   , model_year, origin))
model1 <- lm(log.weight.~log.cylinders., data = cars_log2)
summary(model1)
```

```
##
## Call:
## lm(formula = log.weight. ~ log.cylinders., data = cars_log2)
##
## 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

```
model2 <- lm (log.mpg.~ log.weight.+ log.acceleration.+ model_year+factor(origin)
              , data=cars_log2)
summary(model2)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = cars_log2)
##
## 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
```

b. What is the indirect effect of cylinders on mpg?

```
model1$coefficients[2]*model2$coefficients[2]
```

```
## log.cylinders.
##     -0.7189275
```

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

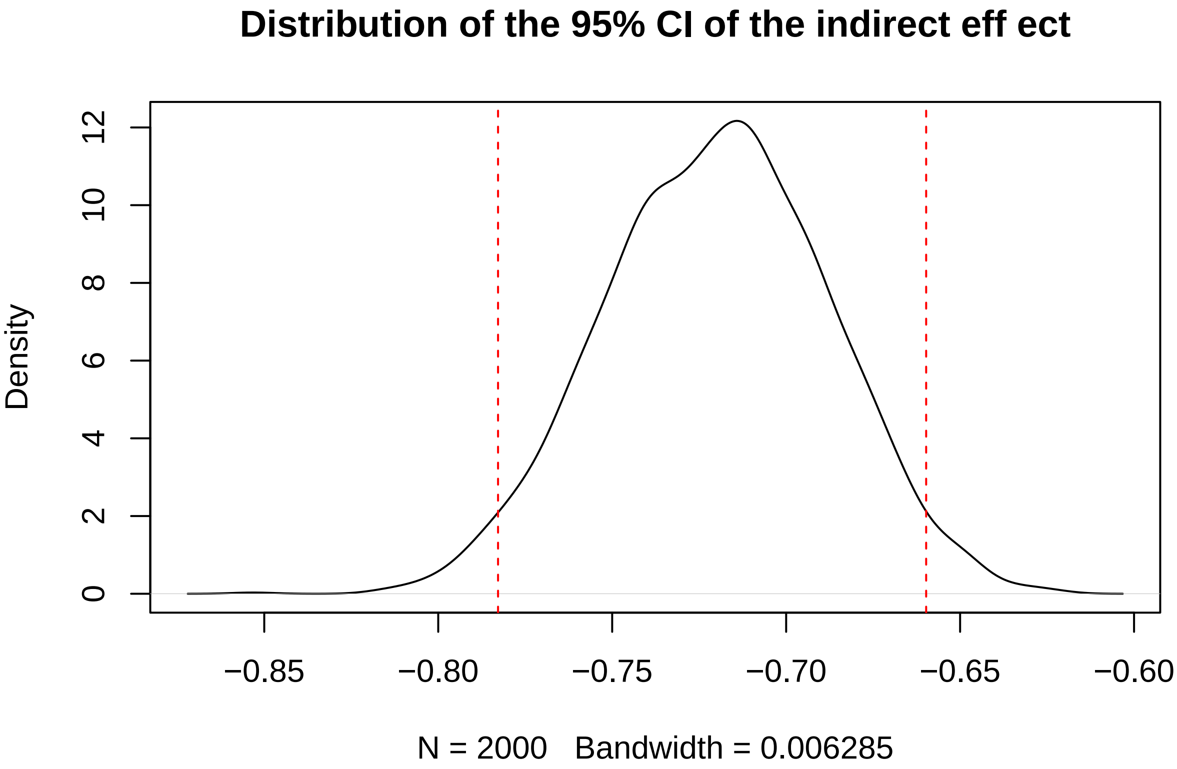
(i) Bootstrap regression models 1 & 2, and compute the indirect effect each time: What is its 95% CI of the indirect effect of log.cylinders. on log.mpg.?

```
boot_mediation <- function(model1,model2,dataset){
  boot_index <- sample(1:nrow(dataset),replace = T)
  data_boot <- dataset[boot_index,]
  regr1 <- lm(model1,data_boot)
  regr2 <- lm(model2,data_boot)
  return(regr1$coefficients[2] * regr2$coefficients[2])
}
indirect <- replicate(2000,boot_mediation(model1,model2,cars_log2))
quantile(indirect,probs = c(0.025,0.975))
```

```
##       2.5%      97.5%
## -0.7828086 -0.6597582
```

(iii) Show a density plot of the distribution of the 95% CI of the indirect effect

```
plot(density(indirect),main = "Distribution of the 95% CI of the indirect eff ect")
abline(v=quantile(indirect,probs = c(0.025,0.975)),lty=2,col="red")
```


Distribution of the 95% CI of the indirect eff ect
Density
0
2
4
6
8
10
12
−0.85
−0.80
−0.75
−0.70
−0.65
−0.60
N = 2000   Bandwidth = 0.006285