BACS HW13

109062710

5/19/2021

Prepare the data set

```
cars_log <- with(Auto, data.frame(log(mpg), log(cylinders), log(displacement),
log(horsepower), log(weight), log(acceleration), year, origin, name))
weight_mean <- mean(cars_log$weight)
```

```
## Warning in mean.default(cars_log$weight): argument is not numeric or logical:
## returning NA
```

```
names(cars_log) <- names(Auto)
head(cars_log)
```

```
##        mpg cylinders displacement horsepower   weight acceleration year origin
## 1 2.890372  2.079442     5.726848   4.867534 8.161660     2.484907   70      1
## 2 2.708050  2.079442     5.857933   5.105945 8.214194     2.442347   70      1
## 3 2.890372  2.079442     5.762051   5.010635 8.142063     2.397895   70      1
## 4 2.772589  2.079442     5.717028   5.010635 8.141190     2.484907   70      1
## 5 2.833213  2.079442     5.710427   4.941642 8.145840     2.351375   70      1
## 6 2.708050  2.079442     6.061457   5.288267 8.375860     2.302585   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

Convert the numbers in `origin` column into names, namely 1 for USA, 2 for Europe, and 3 for Japan.

```
origins <- c("USA", "Europe", "Japan")
cars_log$origin <- factor(cars_log$origin, labels = origins)
```

Question 1

Visualization

i. Split the data set into lightweight cars and heavyweight cars

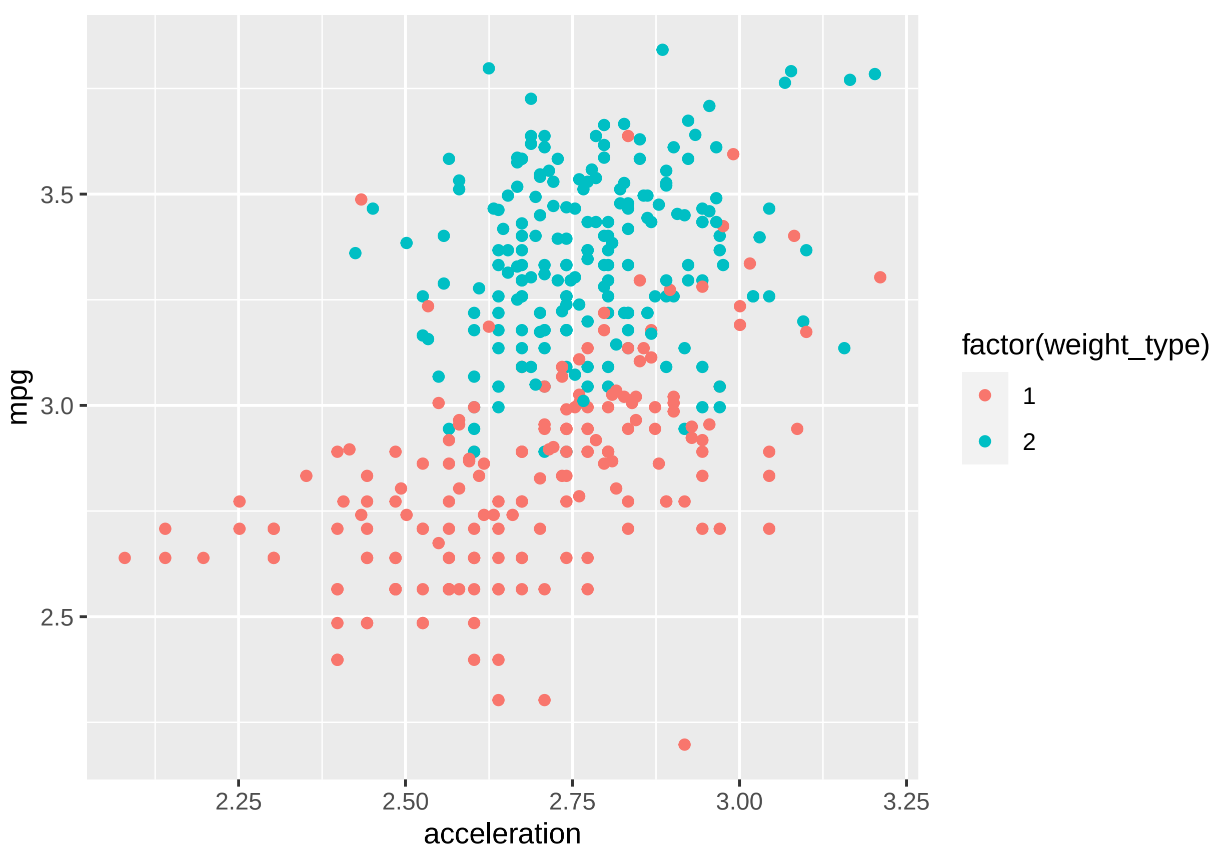
A new column will be made called `weight_type` with `1` as a heavy car and `2` as a light car.

```r
cars_log <- cars_log %>% mutate(weight_type = ifelse(weight >= mean(weight), 1, 2))
head(cars_log)
```

```
##        mpg cylinders displacement horsepower   weight acceleration year origin
## 1 2.890372  2.079442     5.726848   4.867534 8.161660     2.484907   70    USA
## 2 2.708050  2.079442     5.857933   5.105945 8.214194     2.442347   70    USA
## 3 2.890372  2.079442     5.762051   5.010635 8.142063     2.397895   70    USA
## 4 2.772589  2.079442     5.717028   5.010635 8.141190     2.484907   70    USA
## 5 2.833213  2.079442     5.710427   4.941642 8.145840     2.351375   70    USA
## 6 2.708050  2.079442     6.061457   5.288267 8.375860     2.302585   70    USA
##                        name weight_type
## 1 chevrolet chevelle malibu           1
## 2         buick skylark 320           1
## 3        plymouth satellite           1
## 4             amc rebel sst           1
## 5               ford torino           1
## 6          ford galaxie 500           1
```

ii. Create a scatter plot of mpg vs acceleration

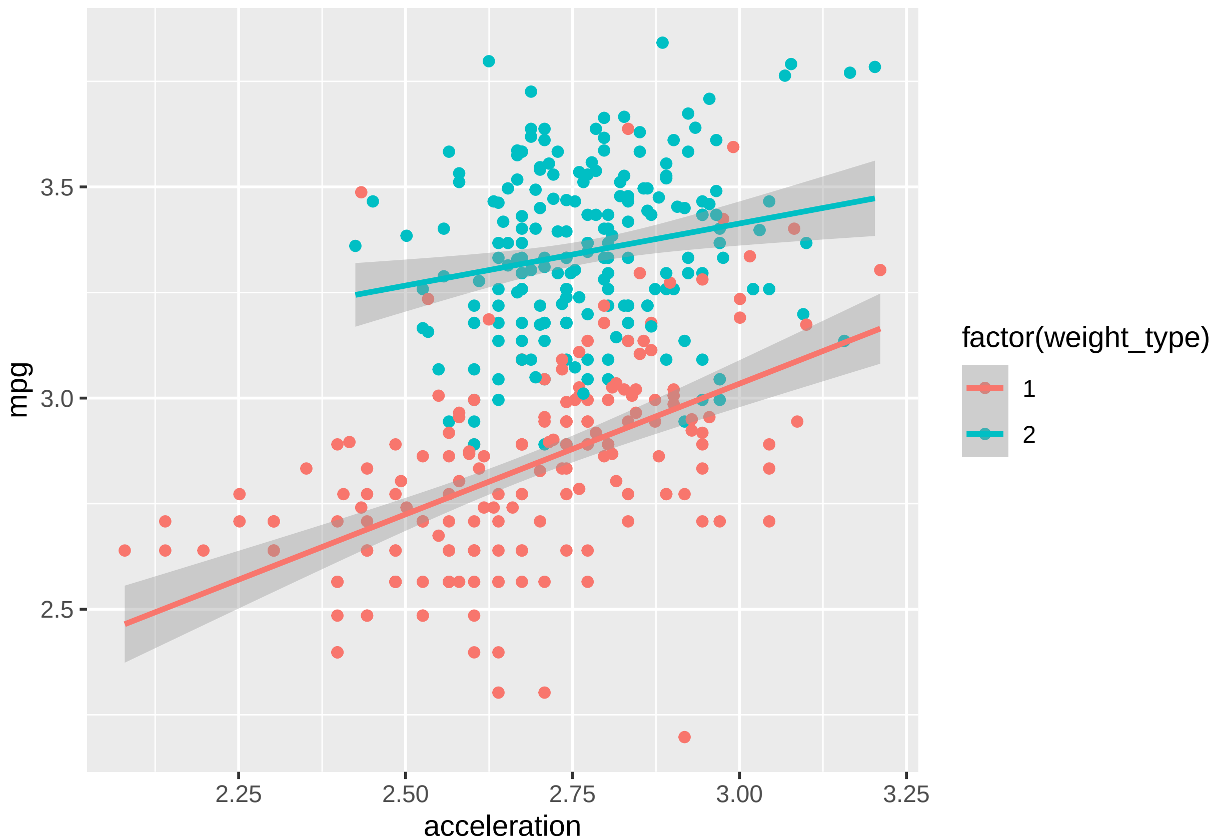
```r
ggplot(data = cars_log, aes(x = acceleration, y = mpg, col = factor(weight_type))) +
    geom_point()
```



iii. Make two seperate regression lines

```
ggplot(data = cars_log, aes(x = acceleration, y = mpg, col = factor(weight_type))) +
    geom_point() +
    geom_smooth(method=lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



b. Report full summaries of light cars and heavy cars

```
light_cars <- cars_log[cars_log$weight < mean(cars_log$weight), ]
light_cars_lm <- lm(mpg ~ weight + acceleration + year + origin, data = light_cars)
summary(light_cars_lm)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + year + origin, data = light_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36684 -0.06688  0.00620  0.06448  0.31576
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.817512   0.606080  11.249   <2e-16 ***
## weight       -0.820783   0.066717 -12.302   <2e-16 ***
## acceleration  0.111434   0.058800   1.895   0.0595 .
## year          0.033109   0.002096  15.798   <2e-16 ***
## originEurope  0.039695   0.021455   1.850   0.0658 .
## originJapan   0.020798   0.019458   1.069   0.2864
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1109 on 196 degrees of freedom
## Multiple R-squared:  0.7034, Adjusted R-squared:  0.6958
## F-statistic: 92.97 on 5 and 196 DF,  p-value: < 2.2e-16
```

```r
heavy_cars <- cars_log[cars_log$weight >= mean(cars_log$weight), ]
heavy_cars_lm <- lm(mpg ~ weight + acceleration + year + origin, data = light_cars)
summary(heavy_cars_lm)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + year + origin, data = light_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36684 -0.06688  0.00620  0.06448  0.31576
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.817512   0.606080  11.249   <2e-16 ***
## weight       -0.820783   0.066717 -12.302   <2e-16 ***
## acceleration  0.111434   0.058800   1.895   0.0595 .
## year          0.033109   0.002096  15.798   <2e-16 ***
## originEurope  0.039695   0.021455   1.850   0.0658 .
## originJapan   0.020798   0.019458   1.069   0.2864
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1109 on 196 degrees of freedom
## Multiple R-squared:  0.7034, Adjusted R-squared:  0.6958
## F-statistic: 92.97 on 5 and 196 DF,  p-value: < 2.2e-16
```

c. What do you observe about light vs. heavy cars?

Both light and heavy cars follow the same trend where as the acceleration increases, so as the distance that a car can cover.

Question 2

a. Which is the moderating variable (not graded)?

A moderating variable is a variable that explains the behavior of an independent variable and a dependent variable. In this case, we can see that 'weight' affects 'acceleration', and 'acceleration' affects 'mpg'. Clearly, 'acceleration' is the moderating variable.

b. Use various regression models

i. Regression without interaction terms

```
summary(
    lm(
        mpg ~ weight +
            acceleration +
            year +
            origin,
        data = cars_log
    )
)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + year + origin, data = cars_log)
##
## 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 ***
## weight       -0.875499   0.029086 -30.101  < 2e-16 ***
## acceleration  0.054377   0.037132   1.464  0.14389
## year          0.032787   0.001731  18.937  < 2e-16 ***
## originEurope  0.056111   0.018241   3.076  0.00225 **
## originJapan   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
```

ii. Regression with an interaction between weight and acceleration

```
summary(
    lm(
        mpg ~
            weight +
            acceleration +
```

```
            year +
            origin +
            weight * acceleration,
        data = cars_log
    )
)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + year + origin + weight *
##     acceleration, data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37795 -0.06904  0.00367  0.06946  0.39735
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.084310   2.780784   0.390  0.69680
## weight               -0.097340   0.341054  -0.285  0.77548
## acceleration          2.357003   1.006243   2.342  0.01967 *
## year                  0.033730   0.001771  19.051  < 2e-16 ***
## originEurope          0.056935   0.018145   3.138  0.00183 **
## originJapan           0.027512   0.018506   1.487  0.13793
## weight:acceleration -0.286724   0.125213  -2.290  0.02257 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1157 on 385 degrees of freedom
## Multiple R-squared:  0.886,  Adjusted R-squared:  0.8843
## F-statistic: 498.9 on 6 and 385 DF,  p-value: < 2.2e-16
```

iii. Regression with a mean-centered interaction term

```
mean_center <- function(data) {
    return(scale(data, center = TRUE, scale = FALSE))
}

summary(
    lm(
        mean_center(mpg) ~
            mean_center(acceleration) +
            mean_center(year)+
            mean_center(Auto$origin) + # Auto$origin is in numeric
            mean_center(mpg * acceleration),
        data = cars_log
    )
)
```

```
##
## Call:
```

```
## lm(formula = mean_center(mpg) ~ mean_center(acceleration) + mean_center(year) +
##     mean_center(Auto$origin) + mean_center(mpg * acceleration),
##     data = cars_log)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.126873 -0.010337  0.005623  0.013265  0.064810
##
## Coefficients:
##                                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                    -3.924e-16  1.152e-03    0.000   1.0000
## mean_center(acceleration)      -1.089e+00  1.067e-02 -102.069   <2e-16 ***
## mean_center(year)               1.087e-03  3.903e-04    2.786   0.0056 **
## mean_center(Auto$origin)        3.155e-04  1.771e-03    0.178   0.8587
## mean_center(mpg * acceleration)  3.636e-01  1.950e-03  186.419   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02281 on 387 degrees of freedom
## Multiple R-squared:  0.9955, Adjusted R-squared:  0.9955
## F-statistic: 2.164e+04 on 4 and 387 DF,  p-value: < 2.2e-16
```

iv. Regression with an orthogonalized interaction term

```
weight_acc_inter_lm <- lm((weight * acceleration) ~
                            weight +
                            acceleration +
                            year + origin,
                        data = cars_log)
cor(weight_acc_inter_lm$residuals, cars_log$weight)
```

```
## [1] 2.617096e-16
```

```
cor(weight_acc_inter_lm$residuals, cars_log$acceleration)
```

```
## [1] -1.374861e-16
```

We can see that both weight and acceleration are orthogonal to each other.

Then we show the linear model summary

```
summary(
    lm(
        mpg ~ weight + acceleration + year + origin + weight_acc_inter_lm$residual,
        data = cars_log
    )
)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + year + origin + weight_acc_inter_lm$residual,
```

```
##     data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37795 -0.06904  0.00367  0.06946  0.39735
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.410974   0.315079  23.521  < 2e-16 ***
## weight                       -0.875499   0.028927 -30.266  < 2e-16 ***
## acceleration                  0.054377   0.036929   1.472  0.14171
## year                          0.032787   0.001722  19.041  < 2e-16 ***
## originEurope                  0.056111   0.018141   3.093  0.00213 **
## originJapan                   0.031937   0.018405   1.735  0.08350 .
## weight_acc_inter_lm$residual -0.286724   0.125213  -2.290  0.02257 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1157 on 385 degrees of freedom
## Multiple R-squared:  0.886,  Adjusted R-squared:  0.8843
## F-statistic: 498.9 on 6 and 385 DF,  p-value: < 2.2e-16
```

c. What is the correlation between the interaction term and the two variables that are multiplied together?

```
# without interaction
no_interaction_weight <- cor((cars_log$weight * cars_log$acceleration), cars_log$weight)
no_interaction_acceleration = cor((cars_log$weight * cars_log$acceleration), cars_log$acceleration)

# mean-centered weight and acceleration
mean_centered_weight <- cor(mean_center(cars_log$weight) * mean_center(cars_log$acceleration), mean_cen
mean_centered_acceleration = cor(mean_center(cars_log$weight) * mean_center(cars_log$acceleration), mean

# orthogonalized weight and acceleration
orthogonalized_weight <- cor(weight_acc_inter_lm$residuals, cars_log$weight)
orthogonalized_acceleration = cor(weight_acc_inter_lm$residuals, cars_log$acceleration)

correlation_matrix <- matrix(
    c(
        no_interaction_weight,
        no_interaction_acceleration,
        mean_centered_weight,
        mean_centered_acceleration,
        orthogonalized_weight,
        orthogonalized_acceleration
    ),
    ncol = 2, byrow=TRUE
)
```

```
correlation_matrix
```

```
##                [,1]          [,2]
```

```
## [1,]  1.090314e-01  8.528828e-01
## [2,] -1.977815e-01  3.445721e-01
## [3,]  2.617096e-16 -1.374861e-16
```

```
rownames(correlation_matrix) <- c("without interaction", "mean-centered", "orthogonalized")
colnames(correlation_matrix) <- c("weight", "acceleration")
round(correlation_matrix, 5)
```

```
##                       weight acceleration
## without interaction  0.10903      0.85288
## mean-centered       -0.19778      0.34457
## orthogonalized       0.00000      0.00000
```