BACS HW10
Student ID: 110065508

Due: 2022/04/24

Question 1. First, use the `R` command

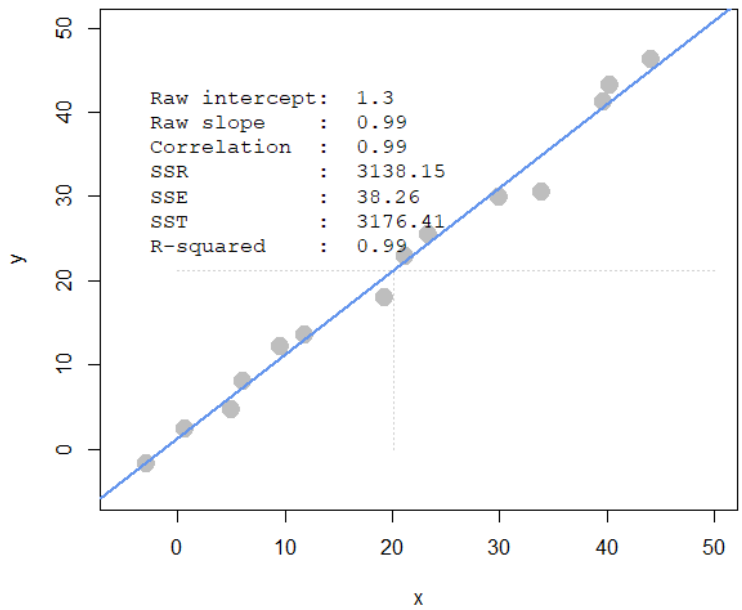
`> source("demo_simple_regression_rsq.R")`

then, call the function

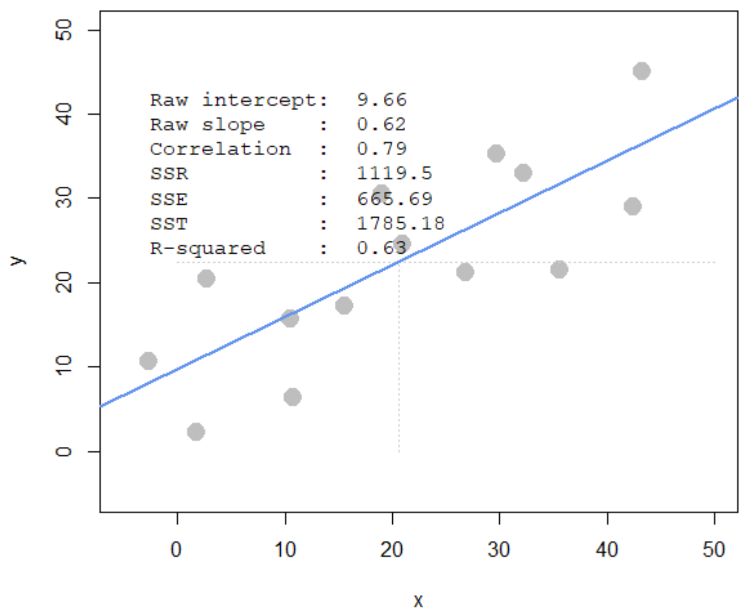
`> interactive_regression_rsq()`

four times to generate four scenarios.

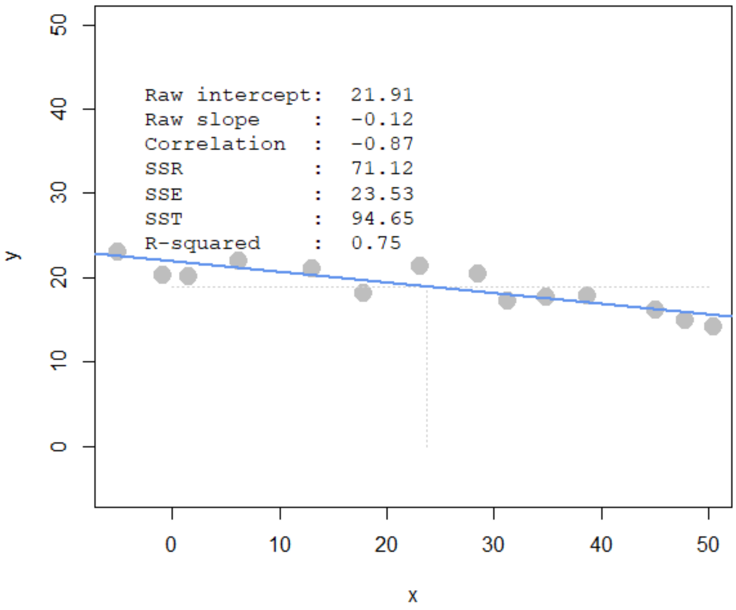
Scenario 1:



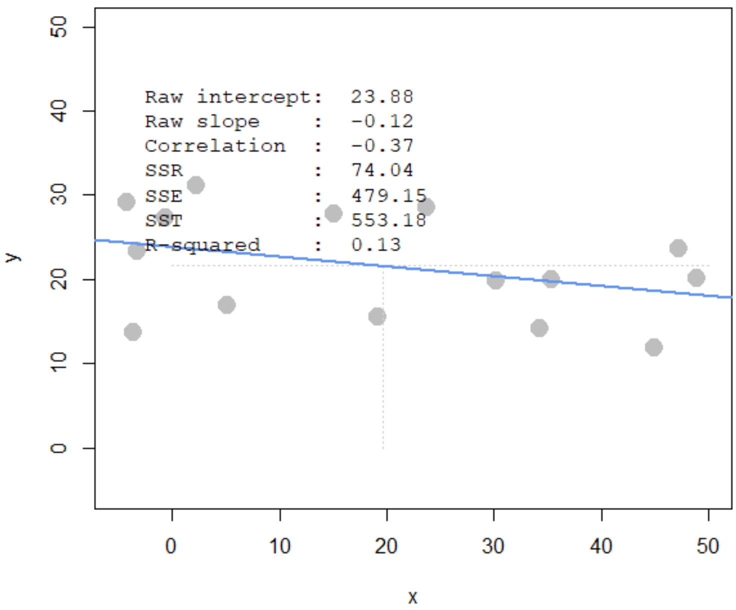
Scenario 2:



Scenario 3:



Scenario 4:



(a) Scenario 1 has a stronger R^2.

(b) Scenario 3 has a stronger R^2.

(c) SSE: Scenario 2 > Scenario 1. SSR: Scenario 1 > Scenario 2. SST: Scenario 1 > Scenario 2.

(d) SSE: Scenario 4 > Scenario 4. SSR: Scenario 4 > Scenario 4. SST: Scenario 4 > Scenario 4.

Question 2. (a) First,read the dataset and directly do the linear regression via `lm()` function..

```
# Question 2 (a)
salary <- read.csv("programmer_salaries.txt", sep="\t")
salary_regression <- lm(salary$Salary ~
                              salary$Experience +
                              salary$Score +
                              salary$Degree) # do linear regression
summary(salary_regression, data=salary)
```

Here's partial results. One can obtain R^2 and the first 5 values of $\hat{y}$ and ϵ here.

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)         7.9448     7.3808   1.076   0.2977
salary$Experience   1.1476     0.2976   3.856   0.0014 **
salary$Score        0.1969     0.0899   2.191   0.0436 *
salary$Degree       2.2804     1.9866   1.148   0.2679
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.396 on 16 degrees of freedom
Multiple R-squared:  0.8468,	Adjusted R-squared:  0.8181
F-statistic: 29.48 on 3 and 16 DF,  p-value: 9.417e-07


> head(salary_regression$fitted.values)
       1        2        3        4        5        6
27.89626 37.95204 26.02901 32.11201 36.34251 38.24380
> head(salary_regression$residuals)
         1          2          3          4          5          6
-3.8962605  5.0479568 -2.3290112  2.1879860 -0.5425072 -0.2437966
```

(b) The hand-craft linear regression code:

```r
# Question 2 (b)
ones <- replicate(length(salary$Salary), 1) # create ones column vector
# Combine column vectors to a matrix
X <- cbind(ones, salary$Experience, salary$Score, salary$Degree)
y <- salary$Salary
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y # some linear algebra
y_hat <- X %*% beta_hat # predicted values
res <- y - y_hat # residuals
SSR <- sum((y_hat-mean(y))^2)
SSE <- sum((y - y_hat)^2)
SST <- SSR + SSE
```

(iii) $\hat{beta} = (7.944849, 1.147582, 0.196937, 2.280424)^T$.

(iv) Use the command

`> head(y_hat)` and `> head(y_hat)`

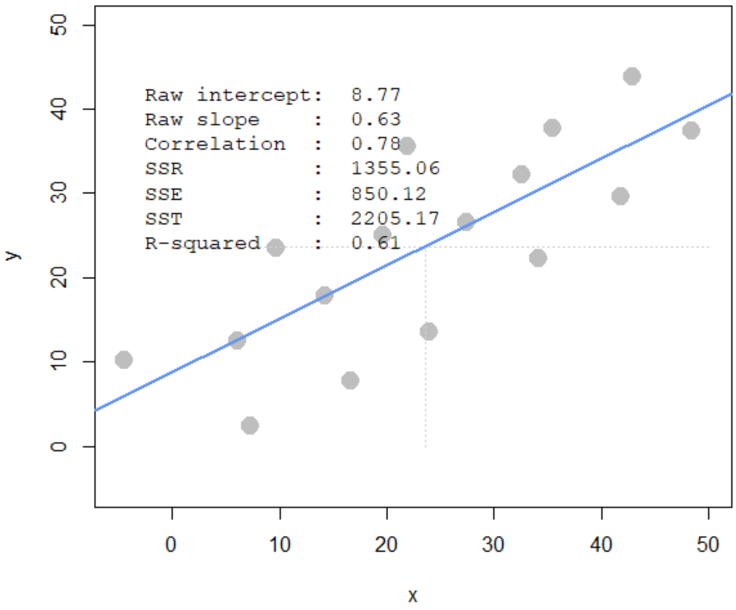
Here's the result:

```
> head(y_hat)   > head(res)
         [,1]             [,1]
[1,] 27.89626   [1,] -3.8962605
[2,] 37.95204   [2,]  5.0479568
[3,] 26.02901   [3,] -2.3290112
[4,] 32.11201   [4,]  2.1879860
[5,] 36.34251   [5,] -0.5425072
[6,] 38.24380   [6,] -0.2437966
```

(v) SSR = 507.896, SSE = 91.88949, and SST = 599.7855.

(c) I generate another plot of scenario 2.

```r
source("demo_simple_regression_rsq.R")
points <- interactive_regression_rsq()
```



Reuse the code from (b):

```r
ones_c <- replicate(length(points[,1]), 1) # create ones
X_c <- cbind(ones_c, points[,1])
y_c <- points[,2]
beta_hat_c <- solve(t(X_c) %*% X_c) %*% t(X_c) %*% y_c
y_hat_c <- X_c %*% beta_hat_c
res_c <- y_c - y_hat_c
SSR_c <- sum((y_hat_c-mean(y_c))^2)
SSE_c <- sum((y_c - y_hat_c)^2)
SST_c <- SSR_c + SSE_c
R2_c_i <- SSR_c/SST_c
R2_c_ii <- cor(y_c, y_hat_c)^2
```

Both the two methods, say `R2_c_i` and `R2_c_ii` returns $R^2 = 0.6144897$. This meets the conclusion from the plot. ■

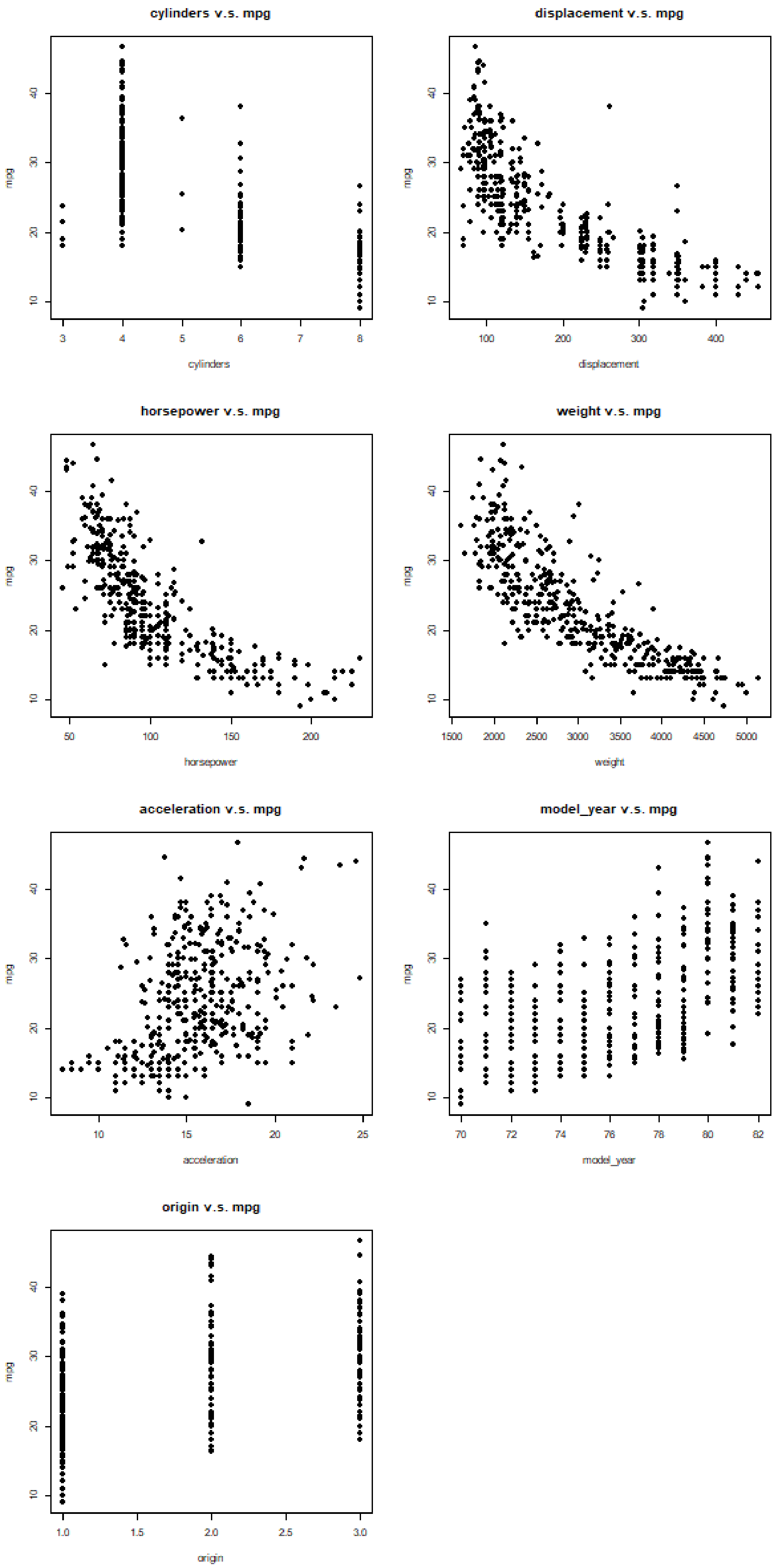
Question 3. Load the data.

```r
# Question 3
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
```

(a)(i) Plot the regression plot to all the other variables expect `car_name`.

```r
# Question 3 (a-i)
png(filename = "3a.png", width = 600, height = 1200) # Subplots
par(mfrow=c(4,2))
cy <- plot(auto$cylinders, auto$mpg, # scatter
           main="cylinders v.s. mpg",
           xlab="cylinders ",
           ylab="mpg",
           pch=19)
di <- plot(auto$displacement, auto$mpg,
           main="displacement v.s. mpg",
```

```
            xlab="displacement",
            ylab="mpg",
            pch=19)
ho <- plot(auto$horsepower, auto$mpg,
            main="horsepower v.s. mpg",
            xlab="horsepower",
            ylab="mpg",
            pch=19)
we <- plot(auto$weight, auto$mpg,
            main="weight v.s. mpg",
            xlab="weight",
            ylab="mpg",
            pch=19)
ac <- plot(auto$acceleration, auto$mpg,
            main="acceleration v.s. mpg",
            xlab="acceleration",
            ylab="mpg",
            pch=19)
my <- plot(auto$model_year, auto$mpg,
            main="model_year v.s. mpg",
            xlab="model_year",
            ylab="mpg",
            pch=19)
or <- plot(auto$origin, auto$mpg,
            main="origin v.s. mpg",
            xlab="origin",
            ylab="mpg",
            pch=19)
dev.off()
```



(ii) Write the correlation matrix to a `.csv` file:

```
# Question 3 (a-ii)
cor_matrix <- cor(auto[,colnames(auto)!="car_name"], # drop column car_name
                  use="pairwise.complete.obs") # omit NA's
```

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin
mpg	1	-0.78	-0.8	-0.78	-0.83	0.42	0.58	0.56
cylinders	-0.78	1	0.95	0.84	0.9	-0.51	-0.35	-0.56
displacement	-0.8	0.95	1	0.9	0.93	-0.54	-0.37	-0.61
horsepower	-0.78	0.84	0.9	1	0.86	-0.69	-0.42	-0.46
weight	-0.83	0.9	0.93	0.86	1	-0.42	-0.31	-0.58
acceleration	0.42	-0.51	-0.54	-0.69	-0.42	1	0.29	0.21
model_year	0.58	-0.35	-0.37	-0.42	-0.31	0.29	1	0.18
origin	0.56	-0.56	-0.61	-0.46	-0.58	0.21	0.18	1

```r
cor_matrix <- round(cor_matrix, digits=2)
write.table(cor_matrix, file="3a.csv")
```

Then by some magic,

I'll answer (iii)-(v) at the same time. First, by (i) and (ii), I found that displacement, horsepower and weight seem to (highly) related to mpg. However these relations seems not linear. Also the variable cylinder has a high correlation ($r = -0.78$) to mpg. However based on the scatter plot, I think it should be viewed as a discrete type data. It is not suit for a linear regression model.

(b) Though some of the variable are not suit for linear regression, I still create a model.

```r
# Question 3 (b)
auto_lr_model <- lm(mpg ~ cylinders+displacement+horsepower+
                    weight+acceleration+model_year+factor(origin), auto)
summary(auto_lr_model)
```

Here's the partial results:

```
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -1.795e+01  4.677e+00  -3.839 0.000145 ***
cylinders       -4.897e-01  3.212e-01  -1.524 0.128215    
displacement     2.398e-02  7.653e-03   3.133 0.001863 ** 
horsepower      -1.818e-02  1.371e-02  -1.326 0.185488    
weight          -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
acceleration     7.910e-02  9.822e-02   0.805 0.421101    
model_year       7.770e-01  5.178e-02  15.005  < 2e-16 ***
factor(origin)2  2.630e+00  5.664e-01   4.643 4.72e-06 ***
factor(origin)3  2.853e+00  5.527e-01   5.162 3.93e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.307 on 383 degrees of freedom
  (因為不存在，6 個觀察量被刪除了)
Multiple R-squared:  0.8242,    Adjusted R-squared:  0.8205 
F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

(i) The variables displacement, weight, model_year, factor(origin)2 and factor(origin)3.

(ii) By (i), based on the plot and the p-values, I believe weight are the most effective at increasing mpg.

(c)(i) Drop the column `car_name` of the dataset and standardize.

```r
# Question 3 (c-i,ii)
auto_std <- data.frame(scale(auto[,colnames(auto)!="car_name"])) # Standardize
auto_lr_model_std <- lm(mpg ~ cylinders+displacement+horsepower+
                        weight+acceleration+
                        model_year, data=auto_std)
summary(auto_lr_model_std)
```

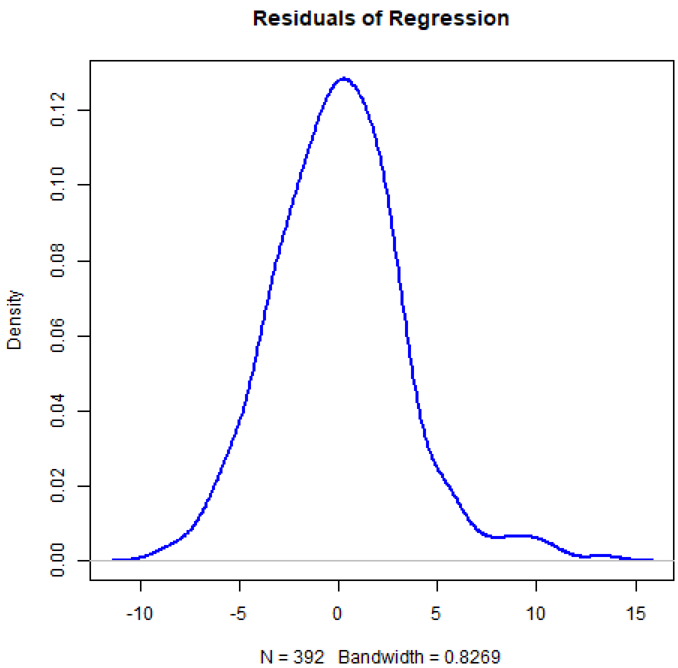
```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.0004236  0.0222112   0.019    0.985    
cylinders    -0.0717877  0.0722763  -0.993    0.321    
displacement  0.1024348  0.0981565   1.044    0.297    
horsepower   -0.0019273  0.0681403  -0.028    0.977    
weight       -0.7361794  0.0725952 -10.141   <2e-16 ***
acceleration  0.0300867  0.0360009   0.836    0.404    
model_year    0.3564069  0.0248929  14.318   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

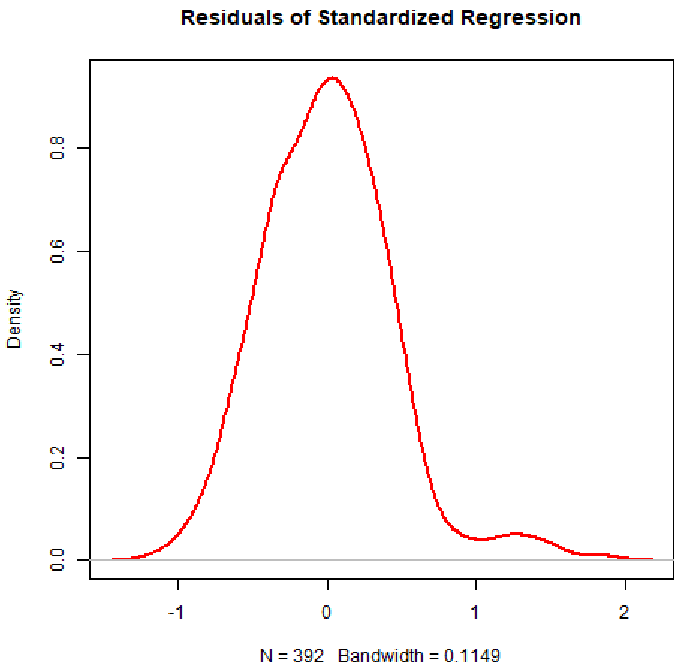
(ii) Based on the result, it's still weight.

(iii) Plot the density of the residuals.

```r
# Question 3 (c-iii)
png(filename = "3c-1.png")
plot(density(auto_lr_model$residuals),
     main="Residuals of Regression",
     col="blue", lwd=2)
dev.off()

png(filename = "3c-2.png")
plot(density(auto_lr_model_std$residuals),
     main="Residuals of Standardized Regression",
     col="red", lwd=2)
dev.off()
```





It looks like in both cases, the residuals are normally distributed and centered around zero, intuitively. ■