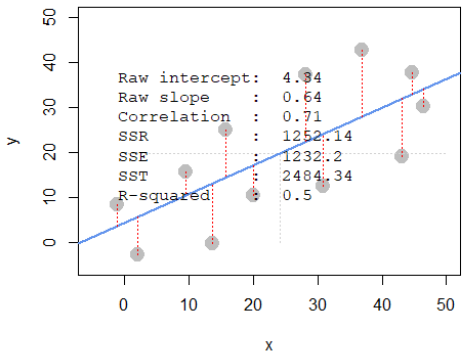
HW_11

i. Plot Scenario 2, storing the returned points: pts <- interactive_regression_rsq()

iii. Add line segments to the plot to show the regression residuals (errors) as follows:

- Get values of y(regression line's estimates of y, given x): y_hat <- regr$fitted.values
- Add segments: segments(ptsx, ptsy, pts$x, y_hat, col="red", lty="dotted")




```
pts <- interactive_regression_rsq()
y_hat <- regr$fitted.values
segments(pts$x, pts$y, pts$x, y_hat, col="red",
lty="dotted")
```

ii.Run a linear model of x and y points to confirm the R2 value reported by the simulation:

```
> regr <- lm(y ~ x, data=pts)
> summary(regr)
Call:
lm(formula = y ~ x, data = pts)

Residuals:
     Min       1Q   Median       3Q      Max
-13.1662  -9.0259   0.5657   6.7317  15.0420

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.3432     5.8038   0.748  0.47149
x             0.6411     0.2011   3.188  0.00969 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.1 on 10 degrees of freedom
Multiple R-squared:  0.504, Adjusted R-squared:  0.4544
F-statistic: 10.16 on 1 and 10 DF,  p-value: 0.009691
```

iv. Use only ptsx, ptsy, y_hat and mean(pts$y) to compute SSE, SSR and SST, and verify R2

```
y_hat <- regr$fitted.values
segments(pts$x, pts$y, pts$x, y_hat, col="red", lty="dotted")
sst <- sum((pts$y - mean(pts$y))^2)
sse <- sum((pts$y - y_hat)^2)
ssr <- sum((y_hat - mean(pts$y))^2)
r_square <- ssr / sst

sst
[1] 2484.345
sse
[1] 1232.2
ssr
[1] 1252.145
r_square
[1] 0.504014
```

b. Comparing scenarios 1 and 2, which do we expect to have a stronger R^2 ?

scenario 1

c. Comparing scenarios 3 and 4, which do we expect to have a stronger R^2 ?

scenario 2

d. Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST?
(do not compute SSE/SSR/SST here – just provide your intuition)

Comparing scenarios 1 and 2,
Scenario 1 has smaller SSR, SSE and SST.

e. Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST?
(do not compute SSE/SSR/SST here – just provide your intuition)

Comparing scenarios 3 and 4,
Scenario 3 has smaller SSR, SSE and SST.

Question 2)

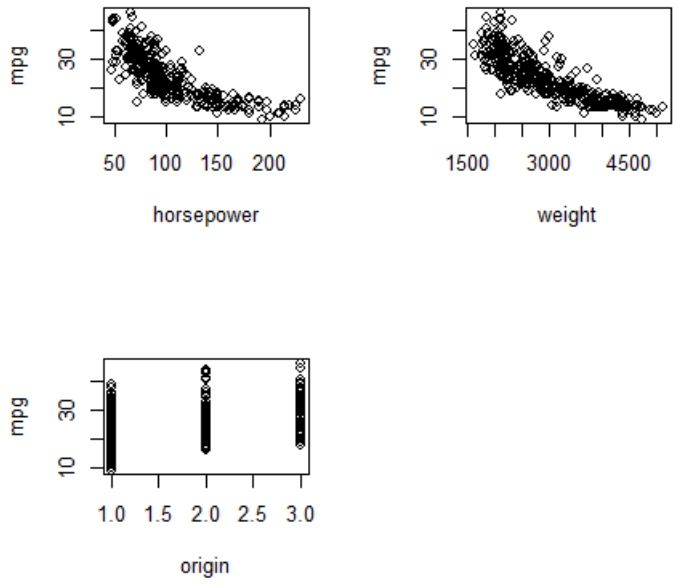
```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
```

a.

i. Visualize the data in any way you feel relevant (report only relevant/interesting ones)

ii. Report a correlation table of all variables, rounding to two decimal places

```
par(mfrow = c(2,2))
plot(x = auto$horsepower, y = auto$mpg, xlab = "horsepower", ylab = "mpg")
plot(x = auto$weight, y = auto$mpg, xlab = "weight", ylab = "mpg")
plot(x = auto$origin, y = auto$mpg, xlab = "origin", ylab = "mpg")
```



iii. From the visualizations and correlations, which variables seem to relate to mpg?

```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")

# auto[, -9]    remove the car type (characters)
cor_matrix <- round(cor(auto[,-9],use="pairwise.complete.obs"), 2)

cor_matrix
                mpg  cylinders  displacement horsepower weight acceleration model_year  origin
mpg            1.00      -0.78         -0.80      -0.78  -0.83         0.42       0.58    0.56
cylinders     -0.78       1.00          0.95       0.84   0.90        -0.51      -0.35   -0.56
displacement  -0.80       0.95          1.00       0.90   0.93        -0.54      -0.37   -0.61
horsepower    -0.78       0.84          0.90       1.00   0.86        -0.69      -0.42   -0.46
weight        -0.83       0.90          0.93       0.86   1.00        -0.42      -0.31   -0.58
acceleration   0.42      -0.51         -0.54      -0.69  -0.42         1.00       0.29    0.21
model_year     0.58      -0.35         -0.37      -0.42  -0.31         0.29       1.00    0.18
origin         0.56      -0.56         -0.61      -0.46  -0.58         0.21       0.18    1.00
```

From the matrix, it seems that cylinder, displacement, horsepower and weight are relating to mpg.

iv. Which relationships might not be linear? (don't worry about linearity for rest of this HW)

Cylinder vs. model year
Displacement vs. model year
Weight vs. model year
Acceleration vs. model year
Acceleration vs. origin
Above pairs are seems not linear.

v. Are there any pairs of independent variables that are highly correlated (r > 0.7)?

```
> abs(cor_matrix) > 0.7
              mpg cylinders displacement horsepower weight acceleration model_year origin
mpg          TRUE      TRUE         TRUE       TRUE   TRUE        FALSE      FALSE  FALSE
cylinders    TRUE      TRUE         TRUE       TRUE   TRUE        FALSE      FALSE  FALSE
displacement TRUE      TRUE         TRUE       TRUE   TRUE        FALSE      FALSE  FALSE
horsepower   TRUE      TRUE         TRUE       TRUE   TRUE        FALSE      FALSE  FALSE
weight       TRUE      TRUE         TRUE       TRUE   TRUE        FALSE      FALSE  FALSE
acceleration FALSE     FALSE        FALSE      FALSE  FALSE        TRUE      FALSE  FALSE
model_year   FALSE     FALSE        FALSE      FALSE  FALSE       FALSE       TRUE  FALSE
origin       FALSE     FALSE        FALSE      FALSE  FALSE       FALSE      FALSE   TRUE
```

From the matrix above,
Mpg vs. cylinder, displacement, horsepower, weight
Cylinder vs. displacement, horsepower, weight
Displacement vs. horsepower, weight
Horsepower vs. weight have highly relation.

```
summary(lm(auto$mpg ~ auto$cylinders + auto$displacement + auto$horsepower + auto$weight +
auto$acceleration + auto$model_year + factor(auto$origin)))
Call:
lm(formula = auto$mpg ~ auto$cylinders + auto$displacement +
    auto$horsepower + auto$weight + auto$acceleration + auto$model_year +
    factor(auto$origin))
Residuals:
     Min       1Q   Median       3Q      Max
-9.0095 -2.0785 -0.0982   1.9856 13.3608
Coefficients:
                        Estimate      Std. Error    t value   Pr(>|t|)
(Intercept)           -1.795e+01   4.677e+00   -3.839   0.000145 ***
auto$cylinders        -4.897e-01   3.212e-01   -1.524   0.128215
auto$displacement      2.398e-02   7.653e-03    3.133   0.001863 **
auto$horsepower       -1.818e-02   1.371e-02   -1.326   0.185488
auto$weight           -6.710e-03   6.551e-04  -10.243   < 2e-16 ***
auto$acceleration      7.910e-02   9.822e-02    0.805   0.421101
auto$model_year        7.770e-01   5.178e-02   15.005   < 2e-16 ***
factor(auto$origin)2   2.630e+00   5.664e-01    4.643   4.72e-06 ***
factor(auto$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
  (6 observations deleted due to missingness)
Multiple R-squared:  0.8242,    Adjusted R-squared:  0.8205
F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

b. i. Which independent variables have a 'significant' relationship with mpg at 1% significance?

Displacement, weight, model_year, origin2(Europe) and origin3(Japan) have significant.

ii. Looking at the coefficients, is it possible to determine which independent variables are the most effective at increasing mpg? If so, which ones, and if not, why not? (hint: units!)

From the results above, the most efficient way to increase mpg is buying the car from Europe and Japan. According to the regression, as long as you buying automobile from Europe and Japan, it will increase mpg 2.630 and 2.853 respectively.

c. i. Create fully standardized regression results: are these slopes easier to compare?

```
Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    acceleration + model_year + factor(origin), data = auto_std)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15270 -0.26593 -0.01257  0.25404  1.70942

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      -0.13323    0.03174  -4.198 3.35e-05 ***
cylinders        -0.10658    0.06991  -1.524  0.12821
displacement      0.31989    0.10210   3.133  0.00186 **
horsepower       -0.08955    0.06751  -1.326  0.18549
weight           -0.72705    0.07098 -10.243  < 2e-16 ***
acceleration      0.02791    0.03465   0.805  0.42110
model_year        0.36760    0.02450  15.005  < 2e-16 ***
factor(origin)2   0.33649    0.07247   4.643 4.72e-06 ***
factor(origin)3   0.36505    0.07072   5.162 3.93e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.423 on 383 degrees of freedom
  (6 observations deleted due to missingness)
Multiple R-squared:  0.8242,    Adjusted R-squared:  0.8205
F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

Yes!! After scaling, the slope are easy to compare though.

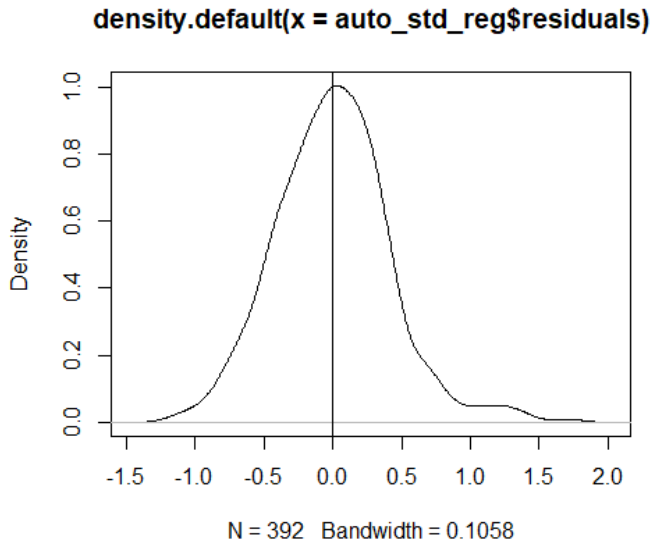
ii. Which ones become significant when we regress mpg over them individually?

summary(lm(mpg ~ cylinders, data = std_auto))
summary(lm(mpg ~ horsepower, data = std_auto))
summary(lm(mpg ~ acceleration, data = std_auto))
After operation, only cylinders become significant when running regression individualy.

```
> summary(lm(mpg ~ cylinders, data = std_auto))

Call:
lm(formula = mpg ~ cylinders, data = std_auto)

Residuals:
      Min       1Q   Median       3Q      Max
-1.82455 -0.43297 -0.08288  0.32674  2.29046

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.834e-15  3.169e-02    0.00        1
cylinders     -7.754e-01  3.173e-02  -24.43   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6323 on 396 degrees of freedom
Multiple R-squared:  0.6012,    Adjusted R-squared:  0.6002
F-statistic: 597.1 on 1 and 396 DF,  p-value: < 2.2e-16
```

iii. Plot the density of the residuals: are they normally distributed and centered around zero?

```
plot(density(auto_std_reg$residuals))
abline(v = mean(auto_std_reg$residuals))
```



It's residual doesn't follow normal distribution and centered around zero.