HW10

109006206

Set Working Directories & Reading Files

```
setwd("/Users/olivia/Documents/Documents/Study/Semester 6/BACS/HW10")
programmer <-read.csv("programmer_salaries.txt", sep="\t")
library(ggplot2)
library(dplyr)
```

QUESTION 1

A) Comparing scenarios 1 and 2, which do we expect to have a stronger R2 ?
Answer : Scenario 1 will have a stronger R^2 , Since data points are narrowly dispersed around a clear and strong trend line , the proportion of variability in y that can be explained by x is likely to be higher. In this case, the linear relationship between x and y is likely to be strong, and the R^2 value will reflect this.

B) Comparing scenarios 3 and 4, which do we expect to have a stronger R2 ?
Answer : Scenario 3 will have a stronger R^2 , Since the data points are narrowly dispersed around a clear trend line, the linear relationship between x and y are strong. This means that the proportion of variability in y that can be explained by x is still relatively high. As a result, the R2 value is likely to be higher. However, it will result in a decreasing manner since most of the data points are around decreasing regression line.

C) Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)
Answer :
Scenario 1 will have a bigger SSR as the line is steep and fits the data closely, capturing most of the variance in the data and bigger SST as the line captures most of the variance in the data.
Scenario 1 will have a smaller SSE as the errors are minimized due to the close fit to the line.
Scenario 2 will have a bigger SSE as the errors are larger due to the greater variability in the data.
Scenario 2 will have a smaller SSR and SST as they have a greater variability.

D) Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)
Answer :
I expect scenario 4 to have a larger SSE and SST a smaller SSR compared to scenario 3.
This is because in Scenario 4, the dispersion of points is wider, meaning there is more variability in the data and therefore a larger amount of error in the model.

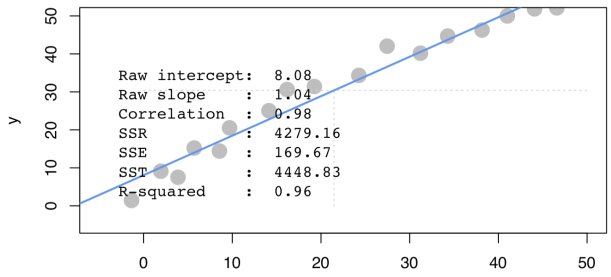


Figure 1: Scenario1

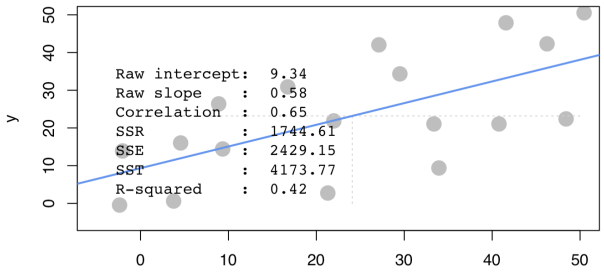


Figure 2: Scenario2

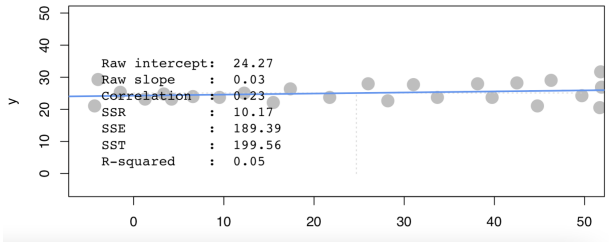


Figure 3: Scenario3

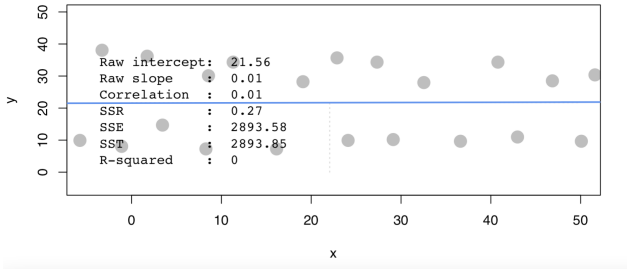


Figure 4: Scenario4

QUESTION 2

A) Use the lm() function to estimate the regression model Salary ~ Experience + Score + Degree. Show the beta coefficients, R2, and the first 5 values of y (*fitted.values*)*and*(residuals)

```
##
## Call:
## lm(formula = Salary ~ Experience + Score + Degree, data = programmer)
##
## Coefficients:
## (Intercept)   Experience        Score       Degree
##      7.9448       1.1476       0.1969       2.2804
```

```
##
## Call:
## lm(formula = Salary ~ Experience + Score, data = programmer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3586 -1.4581 -0.0341  1.1862  4.9102
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.17394    6.15607   0.516  0.61279
## Experience   1.40390    0.19857   7.070 1.88e-06 ***
## Score        0.25089    0.07735   3.243  0.00478 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.419 on 17 degrees of freedom
## Multiple R-squared:  0.8342, Adjusted R-squared:  0.8147
## F-statistic: 42.76 on 2 and 17 DF,  p-value: 2.328e-07
```

```
##
## Call:
## lm(formula = Salary ~ Experience + Score + Degree, data = programmer)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8963 -1.7290 -0.3375  1.9699  5.0480
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9448     7.3808   1.076   0.2977
## Experience    1.1476     0.2976   3.856   0.0014 **
## Score         0.1969     0.0899   2.191   0.0436 *
```

```
## 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
```

```
##        1        2        3        4        5
## 27.89626 37.95204 26.02901 32.11201 36.34251
```

```
##          1          2          3          4          5
## -3.8962605  5.0479568 -2.3290112  2.1879860 -0.5425072
```

B) Use only linear algebra and the geometric view of regression to estimate the regression yourself:

1) Create an X matrix

```
x <- cbind(1, programmer[, c("Experience", "Score", "Degree")])
```

2) Create a y vector with the Salary values

```
y <- programmer$Salary
```

3) Compute the beta_hat vector of estimated regression coefficients

```
beta_hat <- solve(t(x) %*% as.matrix(x)) %*% t(x) %*% y
beta_hat
```

```
##                [,1]
## 1          7.944849
## Experience 1.147582
## Score      0.196937
## Degree     2.280424
```

4) Compute a y_hat vector of estimated y values, and a res vector of residuals

```
y_hat <- as.matrix(x) %*% beta_hat
head(y_hat, 5)
```

```
##          [,1]
## [1,] 27.89626
## [2,] 37.95204
```

```
## [3,] 26.02901
## [4,] 32.11201
## [5,] 36.34251
```

```
res <- y - y_hat
head(res, 5)
```

```
##            [,1]
## [1,] -3.8962605
## [2,]  5.0479568
## [3,] -2.3290112
## [4,]  2.1879860
## [5,] -0.5425072
```

5) Using only the results from (i) – (iv), compute SSR, SSE and SST

```
SSR <- sum((y_hat - mean(y))^2)
SSR
```

```
## [1] 507.896
```

```
SSE <- sum(res^2)
SSE
```

```
## [1] 91.88949
```

```
SST <- sum((y - mean(y))^2)
SST
```

```
## [1] 599.7855
```

C) Compute R^2 for in two ways, and confirm you get the same results:

1) Use any combination of SSR, SSE, and SST

```
R2_1 <- SSR / SST
R2_1
```

```
## [1] 0.8467961
```

2) Use any combination of SSR, SSE, and SST

```
R2_2 <- cor(y, y_hat)^2
R2_2
```

```
##           [,1]
## [1,] 0.8467961
```

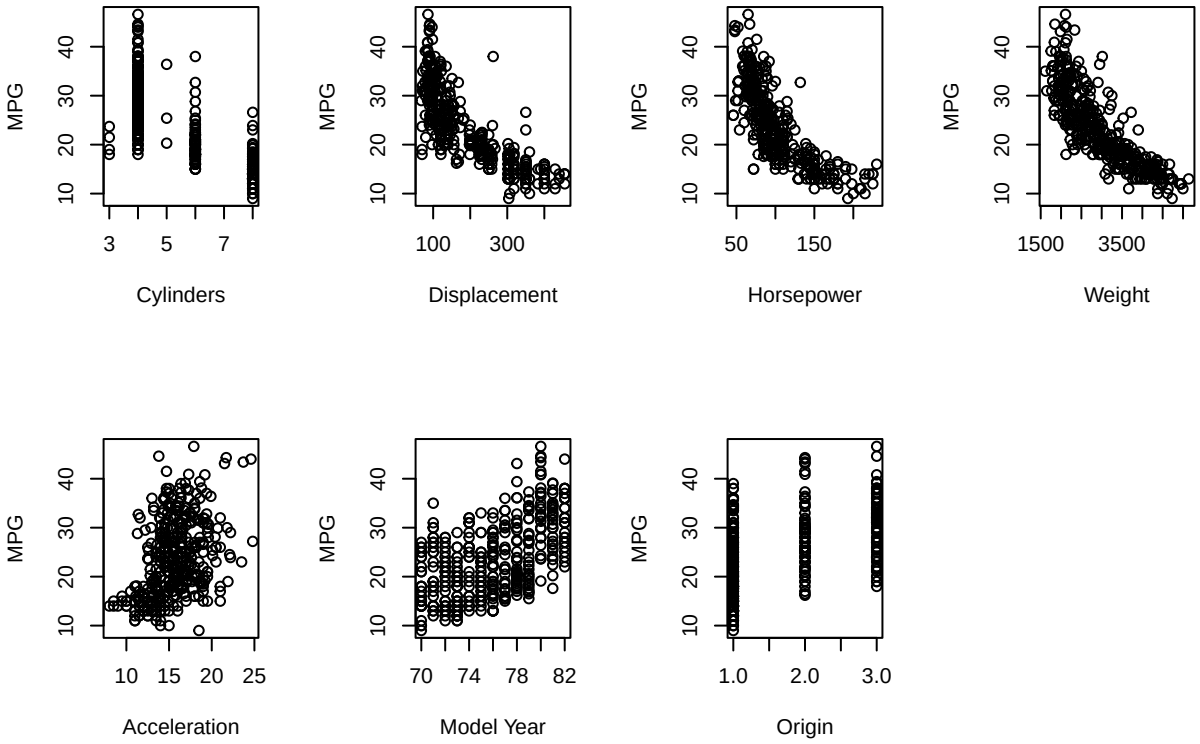
QUESTION 3

```
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) Let's first try exploring this data and problem:

1) Visualize the data as you wish (report only relevant/interesting plots)

```
par(mfrow = c(2, 4))
plot(auto$mpg ~ auto$cylinders, xlab = "Cylinders", ylab = "MPG")
plot(auto$mpg ~ auto$displacement, xlab = "Displacement", ylab = "MPG")
plot(auto$mpg ~ auto$horsepower, xlab = "Horsepower", ylab = "MPG")
plot(auto$mpg ~ auto$weight, xlab = "Weight", ylab = "MPG")
plot(auto$mpg ~ auto$acceleration, xlab = "Acceleration", ylab = "MPG")
plot(auto$mpg ~ auto$model_year, xlab = "Model Year", ylab = "MPG")
plot(auto$mpg ~ auto$origin, xlab = "Origin", ylab = "MPG")
```



2) Report a correlation table of all variables, rounding to two decimal places

```
auto_temp<-auto[1:8]
round(cor(auto_temp,use="pairwise.complete.obs"),2)
```

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

3) From the visualizations and correlations, which variables appear to relate to mpg?

```
auto_temp<-auto[1:8]
round(cor(auto_temp,use="pairwise.complete.obs"),2)
```

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

From the visualizations and correlations, I think cylinders, displacement, horsepower and weight, model year,

and origin appear to relate to mpg

4) Which relationships might not be linear?
Answer: I think relationship of cylinder, acceleration, model year and origin with mpg might not be linear.

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

1. Cylinders and displacement: r = 0.95
2. Weight and displacement: r = 0.93
3. Horsepower and displacement: r = 0.90
4. Weight and cylinders: r = 0.90
5. Horsepower and cylinders: r = 0.84
6. Weight and horsepower: r = 0.86

```
round(cor(auto_temp,use="pairwise.complete.obs"),2)
```

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

B) Let's create a linear regression model where mpg is dependent upon all other suitable variables:

1) Which independent variables have a 'significant' relationship with mpg at 1% significance?

```
autolm <- lm(mpg ~ cylinders + displacement +
               horsepower+ weight+ acceleration +
               model_year+ factor(origin), data = auto)
coefficients_table <- data.frame(summary(autolm)$coefficients)
coefficients_table
```

```
##                       Estimate   Std..Error     t.value     Pr...t..
## (Intercept)     -17.954602067 4.6769339310  -3.8389685 1.445124e-04
## cylinders        -0.489709424 0.3212308567  -1.5244782 1.282146e-01
## displacement      0.023978644 0.0076532690   3.1331244 1.862685e-03
## horsepower       -0.018183464 0.0137085987  -1.3264276 1.854885e-01
## weight           -0.006710384 0.0006551331 -10.2427793 6.375633e-22
## acceleration      0.079103036 0.0982184978   0.8053782 4.211012e-01
## model_year        0.777026939 0.0517840867  15.0051297 2.332943e-40
## factor(origin)2   2.630002360 0.5664146647   4.6432455 4.720373e-06
## factor(origin)3   2.853228228 0.5527363020   5.1620062 3.933208e-07
```

```
significant_variables <- coefficients_table[coefficients_table$`Pr...t..` < 0.01,]
significant_variables
```

```
##                       Estimate   Std..Error    t.value     Pr...t..
## (Intercept)     -17.954602067 4.6769339310  -3.838969 1.445124e-04
## displacement      0.023978644 0.0076532690   3.133124 1.862685e-03
## weight           -0.006710384 0.0006551331 -10.242779 6.375633e-22
## model_year        0.777026939 0.0517840867  15.005130 2.332943e-40
## factor(origin)2   2.630002360 0.5664146647   4.643246 4.720373e-06
## factor(origin)3   2.853228228 0.5527363020   5.162006 3.933208e-07
```

2) 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?
Answer: No, it is not possible to determine which independent variables are the most effective at increasing mpg since all independent variable are not standardized

C) Let's try to resolve some of the issues with our regression model above.

1) Create fully standardized regression results: are these slopes easier to compare?

```
auto_sd <- data.frame(auto[1:7])
auto_sd <- scale(auto_sd)
auto_std <- cbind(auto_sd,auto$origin)
colnames(auto_std)[8] ="origin"
auto_std <- data.frame(auto_std)
```

```
auto_stdregr <- lm(mpg ~ cylinders + displacement + horsepower+ weight+ acceleration + model_year+ facto
summary(auto_stdregr)
```

```
##
## 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
```

2) Regress mpg over each nonsignificant independent variable, individually. Which ones become significant when we regress mpg over them individually?

The non-significant independent variables at the 1% level are Cylinders,Horsepower,Acceleration

```
lm_cylinder <- lm(mpg ~ cylinders, data = auto_std)
summary(lm_cylinder)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders, data = auto_std)
##
## 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
```

```
lm_horsepower <- lm(mpg ~ horsepower, data = auto_std)
summary(lm_horsepower)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = auto_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73632 -0.41699 -0.04395  0.35351  2.16531
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.008784   0.031701  -0.277    0.782
## horsepower  -0.777334   0.031742 -24.489   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6277 on 390 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
lm_acceleration <- lm(mpg ~ acceleration, data = auto_std)
summary(lm_acceleration)
```

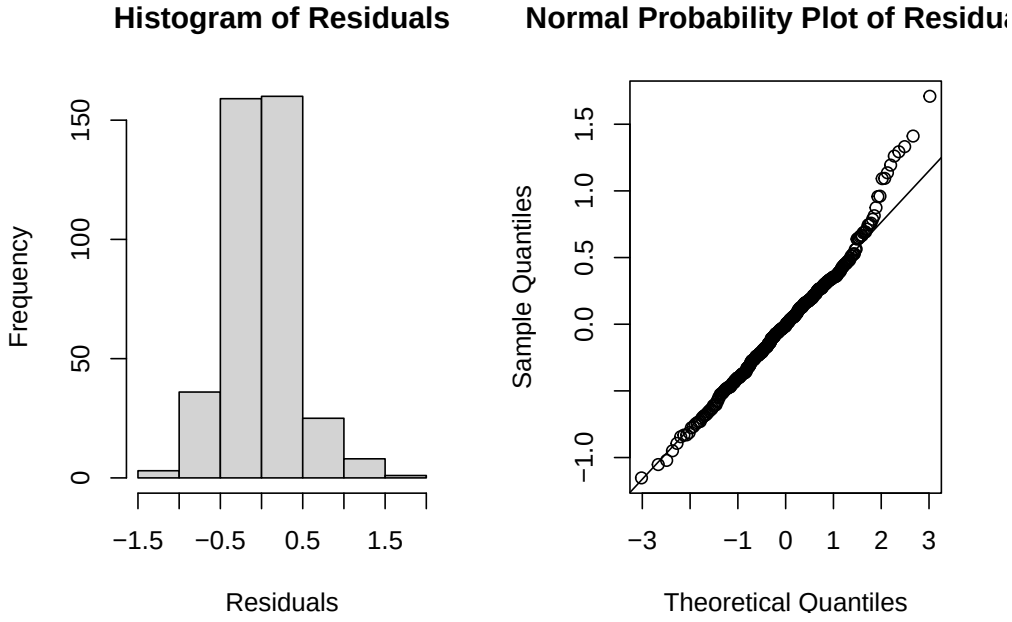
```
##
## Call:
## lm(formula = mpg ~ acceleration, data = auto_std)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.3039 -0.7210 -0.1589  0.6087  2.9672
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.004e-16  4.554e-02   0.000        1
## acceleration 4.203e-01  4.560e-02   9.217   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9085 on 396 degrees of freedom
## Multiple R-squared:  0.1766, Adjusted R-squared:  0.1746
## F-statistic: 84.96 on 1 and 396 DF,  p-value: < 2.2e-16
```

Answer : All of them will become become significant when we regress mpg over them individually

3) Plot the distribution of the residuals: are they normally distributed and centered around zero?

```
par(mfrow = c(1, 2))
hist(auto_stdregr$residuals, main = "Histogram of Residuals", xlab = "Residuals")
qqnorm(auto_stdregr$residuals, main = "Normal Probability Plot of Residuals")
qqline(auto_stdregr$residuals)
```



The data is normally distributed and centered around zero