Student ID: 112077423

```r
library(compstatslib)
library(data.table)
library(ggplot2)
library(tidyr)
library(dplyr)
library(lsa)
```

```
## Warning:     'lsa'          R     4.3.3
```

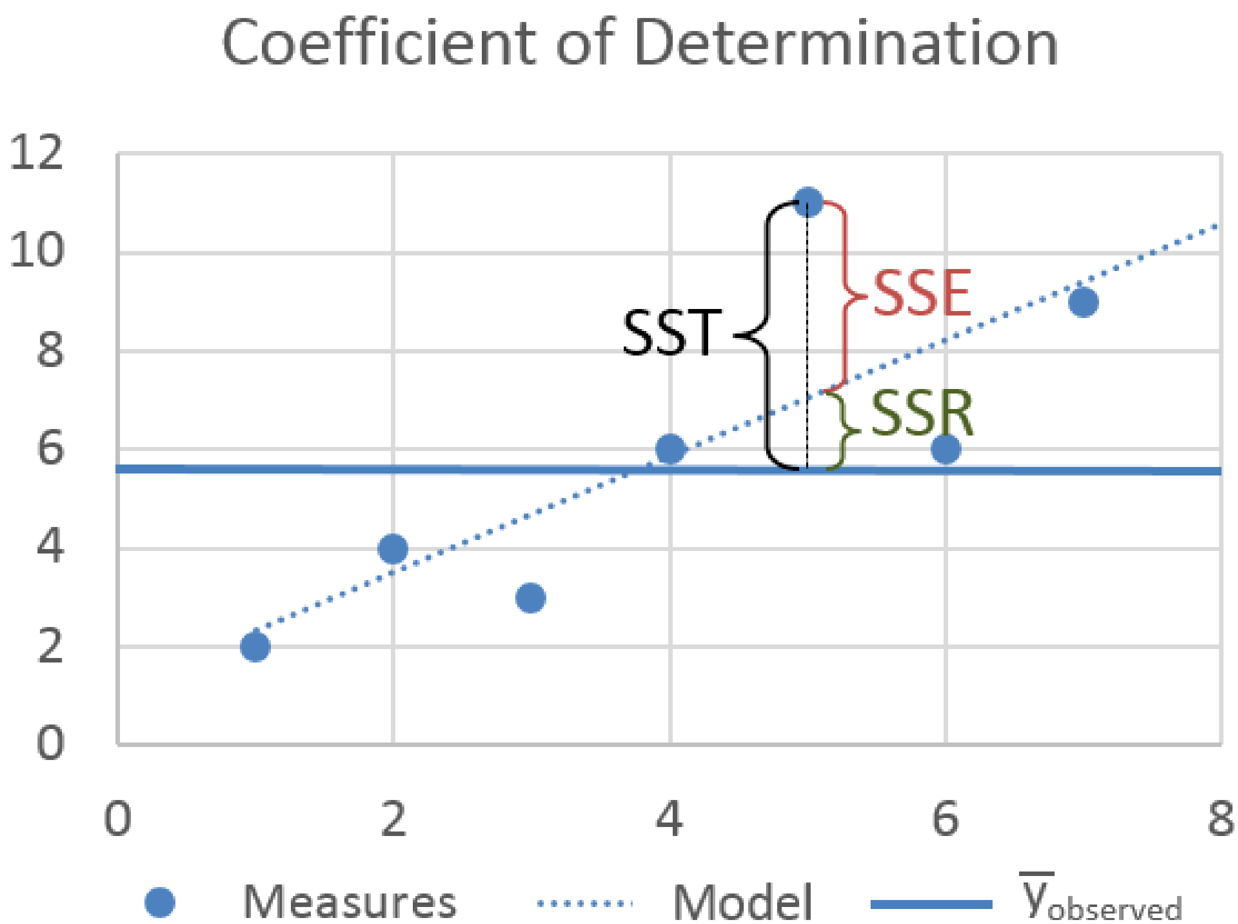
Question 1

(a) Scenario 1 has a stronger R-squared

(b) Scenario 3 has a stronger R-squared

(c) Scenario 1 has a smaller SSE and SST. SSR should be relatively the same for both cases.

(d) Scenario 3 has a smaller SSE and SST. SSR should be relatively the same for both cases.



Question 2(a)

```r
df <- read.csv("programmer_salaries.txt", sep="\t")
head(df)
```

```
##   Experience Score Degree Salary
## 1          4    78      0   24.0
## 2          7   100      1   43.0
## 3          1    86      0   23.7
## 4          5    82      1   34.3
## 5          8    86      1   35.8
## 6         10    84      1   38.0
```

```r
model <- lm(Salary ~ Experience + Score + Degree, data=df)
summary(model)
```

```
##
## Call:
## lm(formula = Salary ~ Experience + Score + Degree, data = df)
##
## 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
```

The beta coefficients:

- Intercept: 7.9448
- Experience: 1.1476
- Score: 0.1969
- Degree: 2.2804

R-squared: 0.8181

```r
out1 <- 'The first 5 fitted values:\t'
out2 <- head(model$fitted.values, 5)
cat(out1, out2, sep=' ')
```

```
## The first 5 fitted values:    27.89626 37.95204 26.02901 32.11201 36.34251
```

```r
cat('\n')
```

```r
out1 <- 'The first 5 residuals:\t\t'
out2 <- head(model$residuals, 5)
cat(out1, out2, sep=' ')
```

```
## The first 5 residuals:        -3.896261 5.047957 -2.329011 2.187986 -0.5425072
```

Question 2(b)

```r
# standardized
X <- cbind(1, df$Experience, df$Score, df$Degree)
y <- df$Salary
```

```r
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
cat('beta_hat:\t', beta_hat)
```

```
## beta_hat:     7.944849 1.147582 0.196937 2.280424
```

```r
y_hat <- X %*% beta_hat
res <- y - y_hat

out1 <- 'The first 5 values of y_hat:\t'
out2 <- head(y_hat, 5)
cat(out1, out2, sep=' ')
```

```
## The first 5 values of y_hat:  27.89626 37.95204 26.02901 32.11201 36.34251
```

```r
cat('\n')
```

```r
out1 <- 'The first 5 residuals:\t\t'
out2 <- head(res, 5)
cat(out1, out2, sep=' ')
```

```
## The first 5 residuals:        -3.896261 5.047957 -2.329011 2.187986 -0.5425072
```

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

out1 <- paste('SSR:\t', SSR)
out2 <- paste('SSE:\t', SSE)
out3 <- paste('SST:\t', SST)
cat(out1, out2, out3, sep='\n')
```

```
## SSR:  507.896013428808
## SSE:  91.8894865712009
## SST:  599.7855
```

Question 2(c)

```r
r2 <- SSR / SST
r2_ <- cor(y, y_hat)^2

out1 <- paste('Method i R-squared:\t', round(r2,2))
out2 <- paste('Method ii R-squared:\t', round(r2_,2))
cat(out1, out2, sep='\n')
```

```
## Method i R-squared:   0.85
## Method ii R-squared:  0.85
```

As we can see, the results are the same.

Question 3(a)

```r
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
# print rows with missing values
print(auto[!complete.cases(auto),])
```

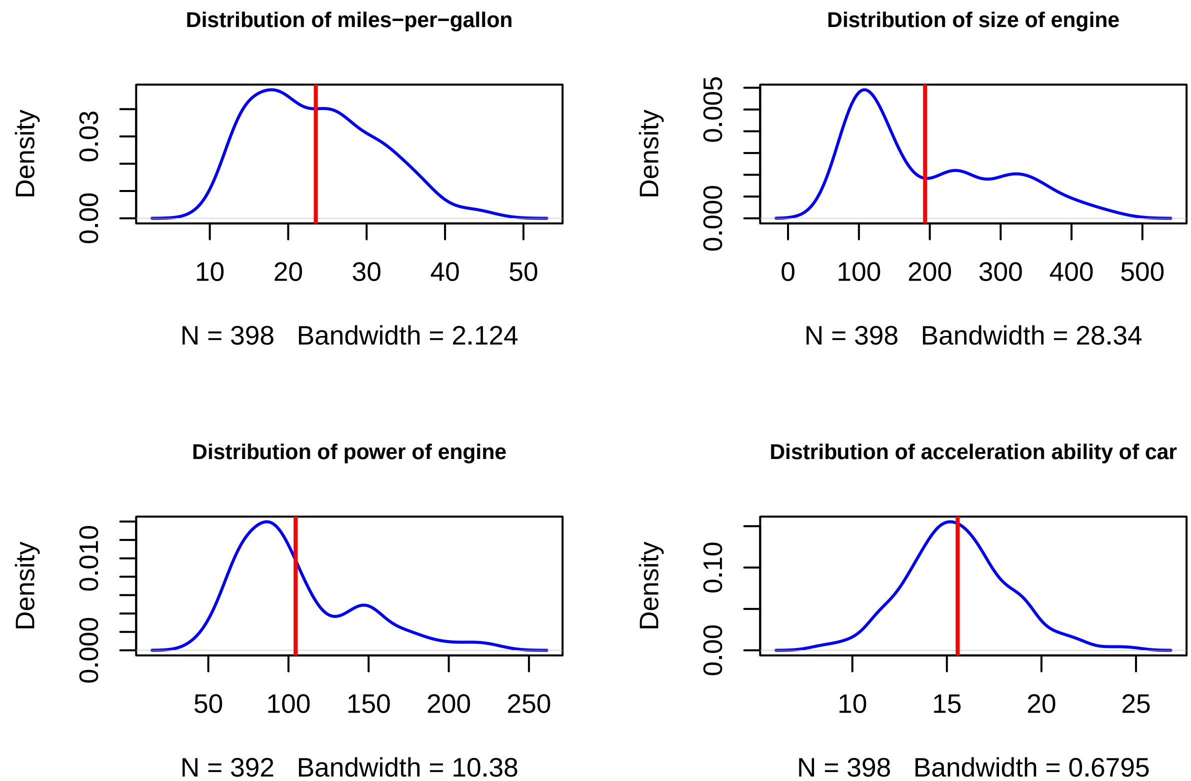
```
##      mpg cylinders displacement horsepower weight acceleration model_year
## 33  25.0         4           98         NA   2046         19.0         71
## 127 21.0         6          200         NA   2875         17.0         74
## 331 40.9         4           85         NA   1835         17.3         80
## 337 23.6         4          140         NA   2905         14.3         80
## 355 34.5         4          100         NA   2320         15.8         81
## 375 23.0         4          151         NA   3035         20.5         82
##     origin             car_name
## 33       1           ford pinto
## 127      1        ford maverick
## 331      2 renault lecar deluxe
## 337      1   ford mustang cobra
## 355      2         renault 18i
## 375      1       amc concord dl
```

```r
# fill missing with the mean
#auto[!complete.cases(auto),'horsepower'] <- mean(auto$horsepower, na.rm = TRUE)
```

(i)

```r
par(mfrow=c(2,2))

plot(density(auto$mpg), col='blue', lwd=1.5,
     main='Distribution of miles-per-gallon', cex.main = 0.8)
abline(v=mean(abs(auto$mpg)), col='red', lw=2)

plot(density(auto$displacement), col='blue', lwd=1.5,
```

```r
     main='Distribution of size of engine', cex.main = 0.8)
abline(v=mean(abs(auto$displacement)), col='red', lw=2)

plot(density(auto$horsepower, na.rm = TRUE), col='blue', lwd=1.5,
     main='Distribution of power of engine', cex.main = 0.8)
abline(v=mean(abs(auto$horsepower), na.rm = TRUE), col='red', lw=2)

plot(density(auto$acceleration), col='blue', lwd=1.5,
     main='Distribution of acceleration ability of car', cex.main = 0.8)
abline(v=mean(abs(auto$acceleration)), col='red', lw=2)
```

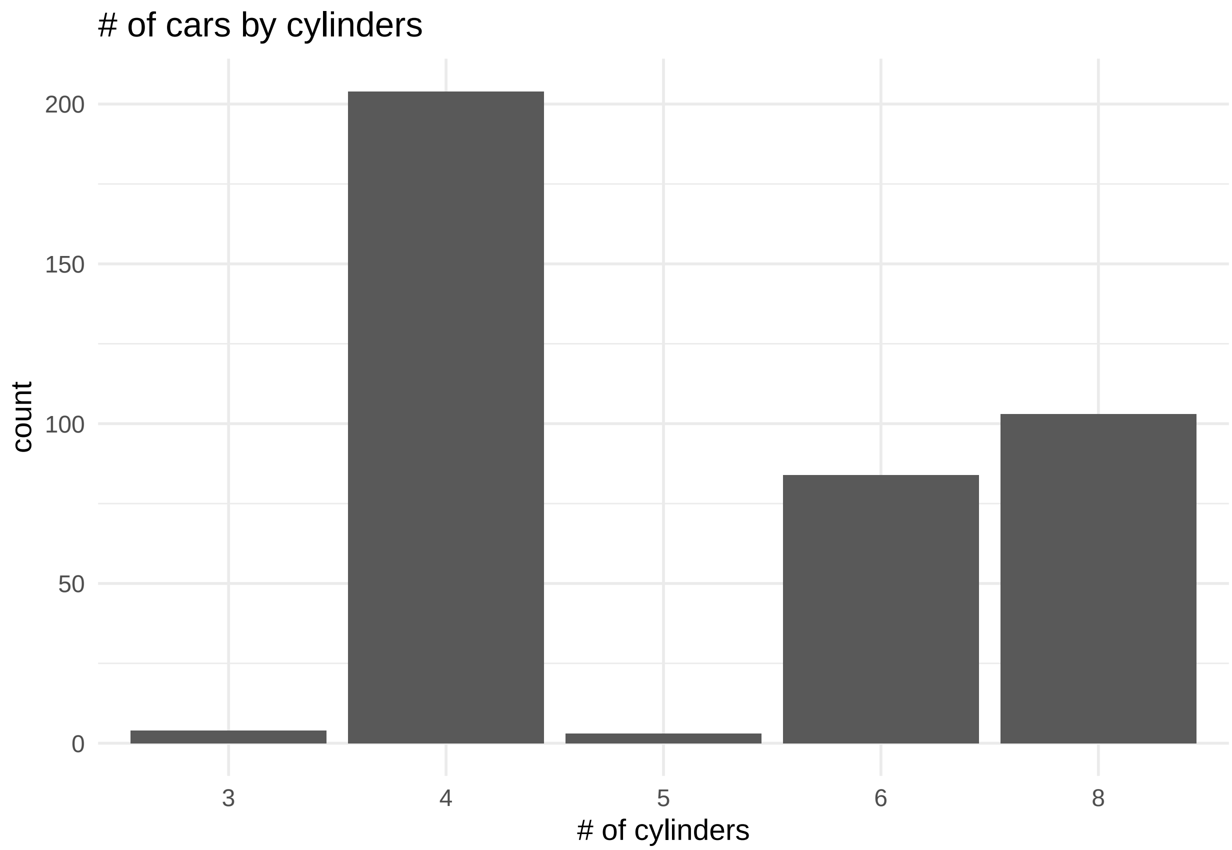


- The mean of miles-per-gallon is around 24. The distribution is skewed to the right;

- The mean of engine size is slightly less than 200. The distribution is skewed to the right;
- The mean of engine power is slightly greater than 100. The distribution is skewed to the right;
- The mean of acceleration ability of car is slightly greater than 15. The distribution is relatively normal.

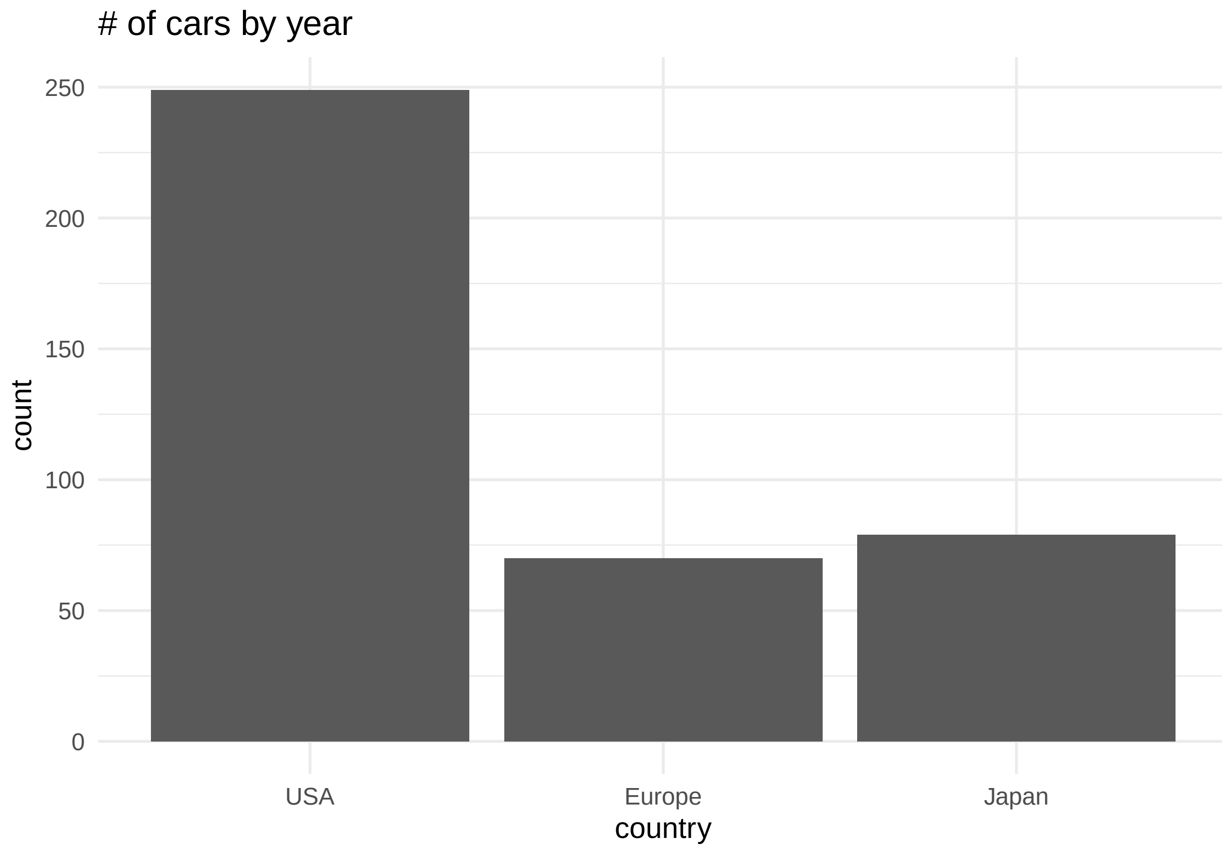
```r
auto_by_cyl <- auto |>
  count(cylinders)

auto_by_cyl$cylinders <- as.factor(auto_by_cyl$cylinders)

ggplot(auto_by_cyl, aes(x = cylinders, y = n)) +
  geom_bar(stat = "identity") +
```

```
  labs(x = '# of cylinders', y = 'count', title = '# of cars by cylinders') +
  theme_minimal()
```



As we can see from the graph, cars with 4 cylinders are the major part of the dataset (>200).

```
pivot <- auto |>
  count(origin)

pivot$origin <- as.factor(pivot$origin)
# 1: USA, 2: Europe, 3: Japan
ggplot(pivot, aes(x = origin, y = n)) +
  geom_bar(stat = "identity") +
  labs(x = 'country', y = 'count', title = '# of cars by year') +
  scale_x_discrete(labels = c('1' = 'USA', '2' = 'Europe', '3' = 'Japan')) +
  theme_minimal()
```



As we can see from the graph, most of the cars are from USA (~250).

(ii)

```
tmp_df <- auto[,1:8]
round(cor(tmp_df,  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
```

(iii)

From visualizations, we can notice that distributions of the size of the engine and power of the engine are similar to the distribution of miles-per-gallon. From the correlation table, mpg/cylinders, mpg/displacement, mpg/horsepower, and mpg/weight have a strong negative correlation. Besides, mpg and model_year are correlated in a sense.

(iv)

model_year/origin and acceleration/origin might not be linear since their correlation is low.

(v)

cylinders, displacement, weight and horsepower are highly correlated (r > 0.7).

Question 3(b)

```r
model <- lm(mpg ~ weight + model_year + factor(origin) +
            cylinders + acceleration + horsepower + displacement, data=auto)
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ weight + model_year + factor(origin) + cylinders +
##     acceleration + horsepower + displacement, data = auto)
##
## 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 ***
## weight          -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## 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 ***
## cylinders       -4.897e-01  3.212e-01  -1.524 0.128215
## acceleration     7.910e-02  9.822e-02   0.805 0.421101
## horsepower      -1.818e-02  1.371e-02  -1.326 0.185488
## displacement     2.398e-02  7.653e-03   3.133 0.001863 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
##   (6                  )
## 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) All the independent variables, except cylinders, acceleration, and horsepower, in the model have a 'significant' relationship with mpg at 1% significance.

(ii)

factor(origin)2 (JP): change in mpg relative to origin 1 (US) factor(origin)3 (EU): change in mpg relative to origin 1 (US)

Question 3(c)

(i)

```r
# no need to standardize origin since it's categorical
model_ <- lm(scale(mpg) ~ scale(weight) + scale(model_year) +
            factor(origin) + scale(cylinders) + scale(acceleration) +
            scale(horsepower) + scale(displacement), data=auto)

summary(model_)
```

```
##
## Call:
## lm(formula = scale(mpg) ~ scale(weight) + scale(model_year) +
##     factor(origin) + scale(cylinders) + scale(acceleration) +
##     scale(horsepower) + scale(displacement), data = auto)
##
## 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 ***
## scale(weight)       -0.72705    0.07098 -10.243  < 2e-16 ***
## scale(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 ***
## scale(cylinders)    -0.10658    0.06991  -1.524  0.12821
## scale(acceleration)  0.02791    0.03465   0.805  0.42110
## scale(horsepower)   -0.08955    0.06751  -1.326  0.18549
## scale(displacement)  0.31989    0.10210   3.133  0.00186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.423 on 383 degrees of freedom
##   (6                     )
## 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 find the results of the standardized version to be easier to interpret.

(ii)

```r
model <- lm(scale(mpg) ~ scale(cylinders), data=auto)
summary(model)
```

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

```r
model <- lm(scale(mpg) ~ scale(acceleration), data=auto)
summary(model)
```

```
##
## Call:
## lm(formula = scale(mpg) ~ scale(acceleration), data = auto)
##
## 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
## scale(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
```

```r
model <- lm(scale(mpg) ~ scale(horsepower), data=auto)
summary(model)
```

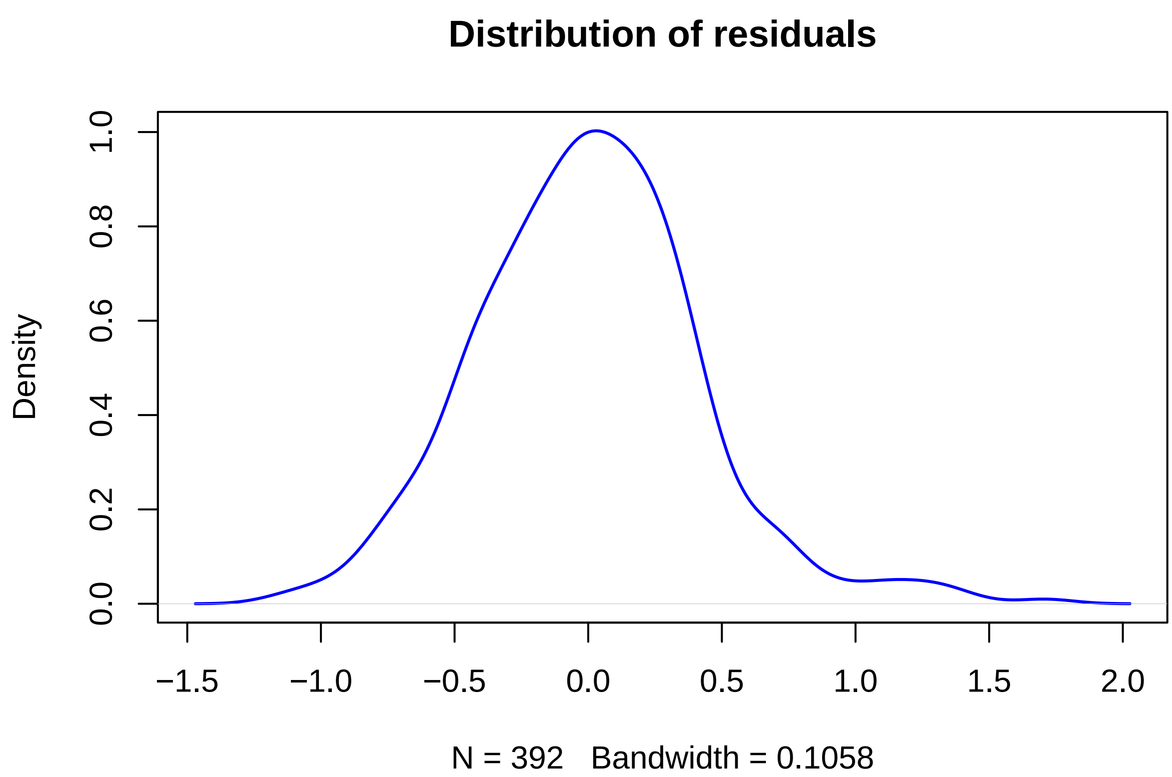
```
##
## Call:
## lm(formula = scale(mpg) ~ scale(horsepower), data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73632 -0.41699 -0.04395  0.35351  2.16531
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.008784   0.031701  -0.277    0.782
## scale(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                   )
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

All three variables become significant when regress mpg over them individually.

(iii)

```r
# get the residuals of standardized model
residuals <- model_$residuals
plot(density(residuals), col='blue', lwd=1.5, main='Distribution of residuals')
```



Looking at the graph, we can say that residuals are normally distributed in a sense and centered around zero.