Business Analytics using Statistical Modeling
Assignment 10

Question 1

Understand each of these four scenarios by simulating them:

Scenario 1: Consider a very narrowly dispersed set of points that have a negative or positive steep slope

Scenario 2: Consider a widely dispersed set of points that have a negative or positive steep slope

Scenario 3: Consider a very narrowly dispersed set of points that have a negative or positive shallow slope

Scenario 4: Consider a widely dispersed set of points that have a negative or positive shallow slope

a. Let's dig into what regression is doing to compute model fit:

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

```
pts <- interactive_regression_rsq()
```

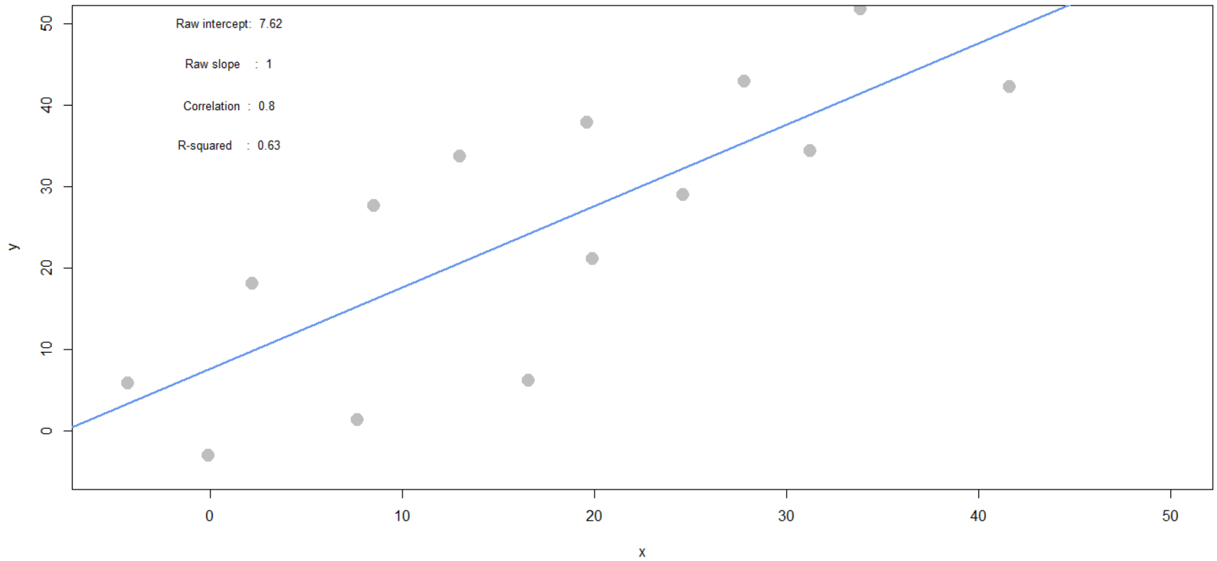


Figure 1: Scenario 2

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 
-18.0671  -6.8187  -0.4291   9.8094  13.1206 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.6201     4.7774   1.595 0.136690    
x             0.9992     0.2199   4.544 0.000673 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.82 on 12 degrees of freedom
Multiple R-squared:  0.6324,    Adjusted R-squared:  0.6018 
F-statistic: 20.65 on 1 and 12 DF,  p-value: 0.0006734
```

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

Get the values of $\hat{y}$ (regression line's estimates of y, given x)

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

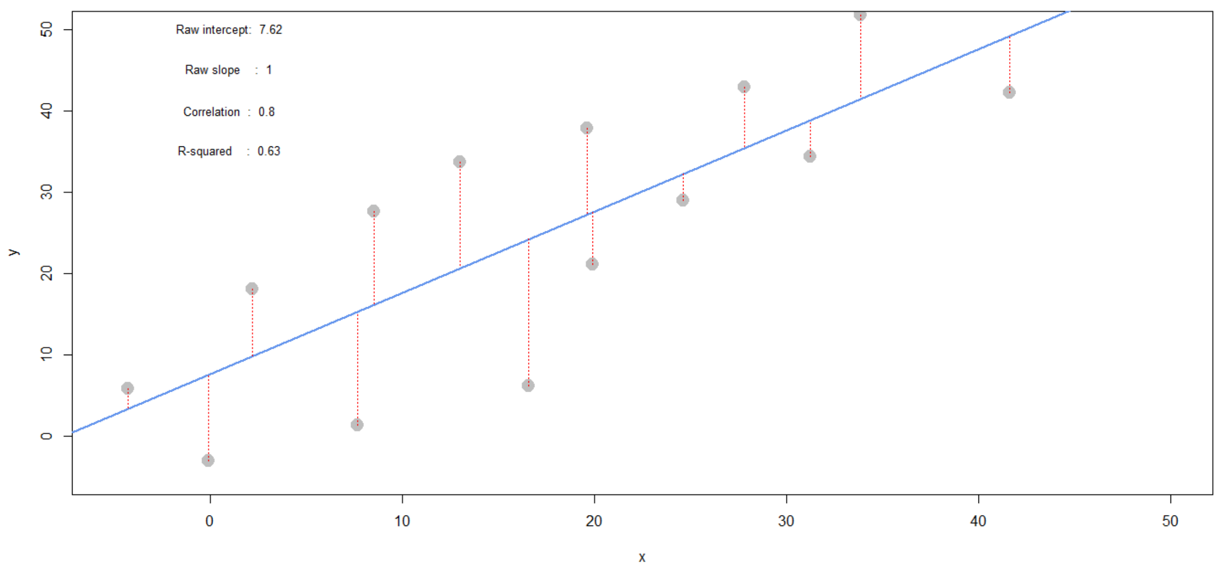


Figure 2: Scenario 2 + line segments to show the regression residuals

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

```
sse <- sum((pts$y - y_hat) ^ 2)
ssr <- sum((y_hat - mean(pts$y)) ^ 2)
sst <- sse + ssr
rsq <- ssr / sst
rsq
[1] 0.632422
```

b. Comparing scenarios 1 and 2, which do we expect to have a stronger R2?

Scenario 1

c. Comparing scenarios 3 and 4, which do we expect to have a stronger R2 ?

Scenario 3

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)

Bigger SSE: Scenario 2

Bigger SSR: Scenario 1

Bigger SST: could be either way

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)

Bigger SSE: Scenario 4

Bigger SSR: could be either way

Bigger SST: Scenario 4

Question 2

Take a look at a data set (auto-data.txt). We are interested in explaining what kind of cars have higher fuel efficiency (measured by mpg).

1. mpg: miles-per-gallon (dependent variable)
2. cyl: cylinders in engine
3. disp: size of engine
4. hp: power of engine
5. w: weight of car
6. acc: acceleration ability of car
7. year: year model was released
8. ori: place car was designed (1: USA, 2: Europe, 3: Japan)
9. name: make and model names

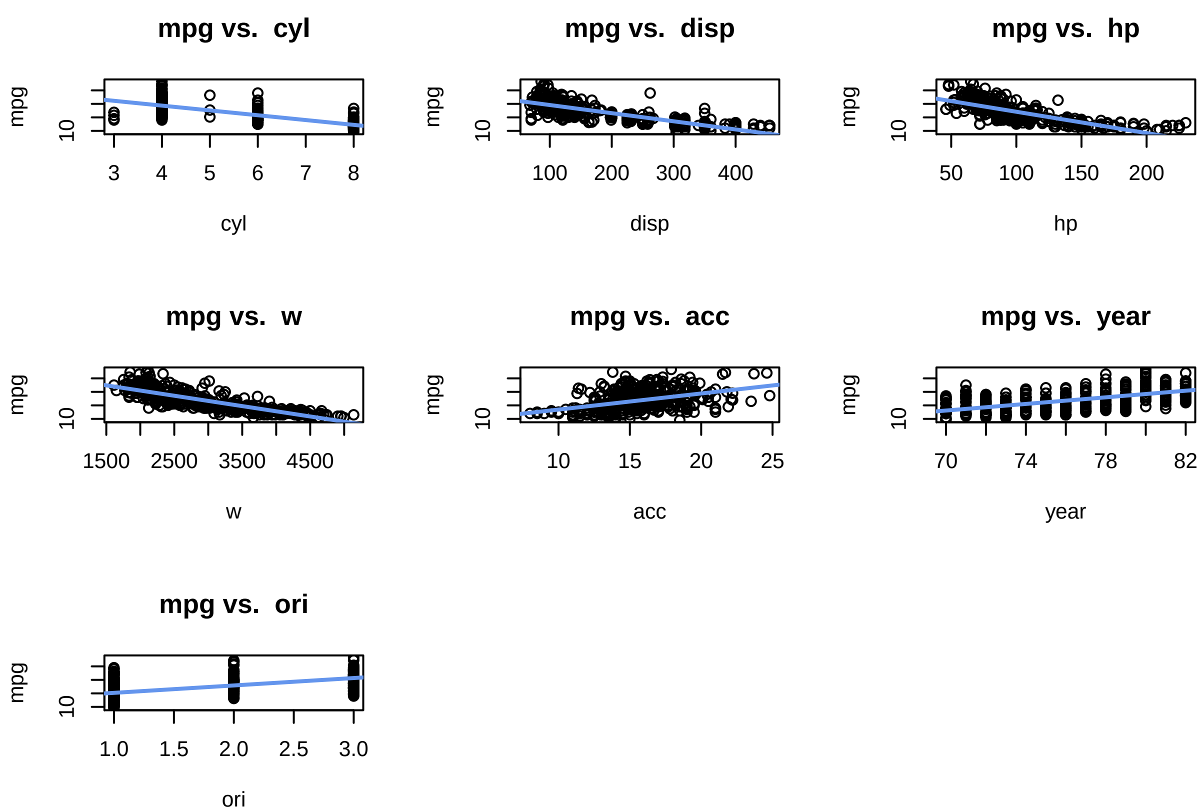
This data set has some missing values ('?' in data set), and it lacks a header row with variable names.

```r
auto <- read.table("../10-auto-data.txt", header = FALSE, na.strings = "?")
names(auto) <- c("mpg", "cyl", "disp", "hp", "w", "acc", "year", "ori", "name")
```

a. Let's first try exploring this data and problem:

i. Visualize the data in any way you feel relevant.

```r
plot_mpg <- function(x) {
  plot(auto[, x], auto$mpg, main = paste('mpg vs. ', names(auto)[x]), ylab = 'mpg',
       xlab = names(auto)[x])
  regr <- lm(auto$mpg ~ auto[, x])
  regr_summary <- summary(regr)
  abline(regr, lwd=2, col = "cornflowerblue")
}
par(mfrow = c(3, 3))
for(i in 2:8) {
  plot_mpg(i)
}
par(mfrow = c(1, 1))
```



ii. Report a correlation table of all variables, rounding to two decimal places (in the cor(. . .) function, set use="pairwise.complete.obs" to handle missing values)

```r
auto_cor <- round(cor(auto[1:8], use = 'pairwise.complete.obs'), 2)
library(pander)
```

```
## Warning: package 'pander' was built under R version 3.2.5
```

```r
pandoc.table(auto_cor, style = 'rmarkdown', justify = 'right', plain.ascii = TRUE)
```

```
##
##
## |            |   mpg |   cyl |   disp |    hp |     w |   acc |   year |   ori |
## |-----------:|------:|------:|-------:|------:|------:|------:|-------:|------:|
## |        mpg |     1 | -0.78 |   -0.8 | -0.78 | -0.83 |  0.42 |   0.58 |  0.56 |
## |        cyl | -0.78 |     1 |   0.95 |  0.84 |   0.9 | -0.51 |  -0.35 | -0.56 |
## |       disp |  -0.8 |  0.95 |      1 |   0.9 |  0.93 | -0.54 |  -0.37 | -0.61 |
## |         hp | -0.78 |  0.84 |    0.9 |     1 |  0.86 | -0.69 |  -0.42 | -0.46 |
## |          w | -0.83 |   0.9 |   0.93 |  0.86 |     1 | -0.42 |  -0.31 | -0.58 |
## |        acc |  0.42 | -0.51 |  -0.54 | -0.69 | -0.42 |     1 |   0.29 |  0.21 |
## |       year |  0.58 | -0.35 |  -0.37 | -0.42 | -0.31 |  0.29 |      1 |  0.18 |
## |        ori |  0.56 | -0.56 |  -0.61 | -0.46 | -0.58 |  0.21 |   0.18 |     1 |
```

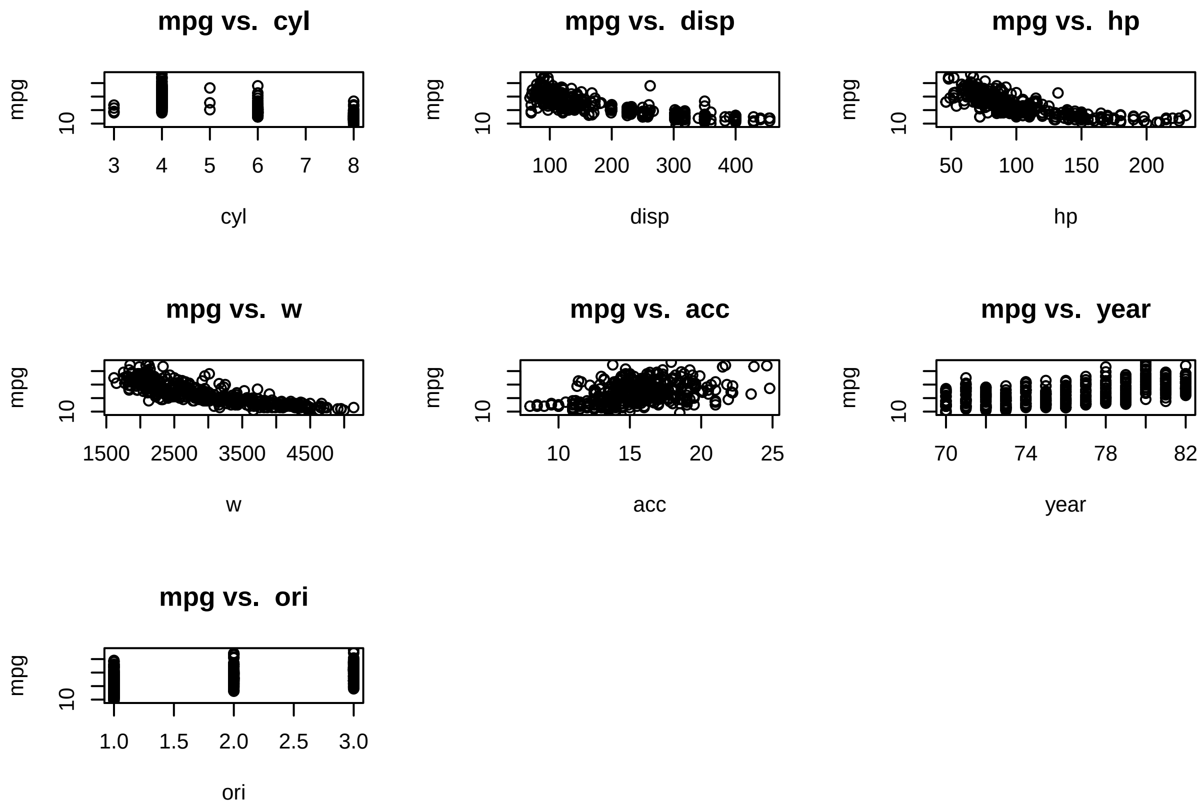
iii. From the visualizations and correlations, which variables seem to relate to mpg?

```r
names(auto[2:8][, which(abs(
  cor(auto$mpg, auto[2:8], use = 'pairwise.complete.obs')) > 0.7)])
```

```
## [1] "cyl"  "disp" "hp"   "w"
```

iv. Which relationships might not be linear?

```
par(mfrow = c(3, 3))
for(i in 2:8) {
  plot(auto[, i], auto$mpg, main = paste('mpg vs. ', names(auto)[i]), ylab = 'mpg',
       xlab = names(auto)[i])
}
par(mfrow = c(1, 1))
```



Seems like `mpg` is not linearly related with `displacement`, `horsepower`, and `weight`.

v. Are any of the independent variables highly correlated (r > 0.7) with others?

```
diag(auto_cor) <- NA
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.2.5
```

```
auto_cor_melt <- melt(auto_cor)
auto_cor_melt <- auto_cor_melt[complete.cases(auto_cor_melt), ]
high_cor <- auto_cor_melt[auto_cor_melt$value > 0.7, ]
high_cor[, 1:2] <- t(apply(high_cor[, 1:2], 1, sort))
high_cor <- high_cor[!duplicated(high_cor), ]
high_cor
```

```
##    Var1 Var2 value
## 11  cyl disp  0.95
## 12  cyl   hp  0.84
## 13  cyl    w  0.90
## 20 disp   hp  0.90
## 21 disp    w  0.93
## 29   hp    w  0.86
```

b. Let's try an ordinary linear regression, where mpg is dependent upon all other suitable variables

```r
regr_mpg <- lm(mpg ~ cyl + disp + hp + w + acc + year + factor(ori), data = auto)
summary(regr_mpg)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + w + acc + year + factor(ori),
##     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 ***
## cyl          -4.897e-01  3.212e-01  -1.524 0.128215
## disp          2.398e-02  7.653e-03   3.133 0.001863 **
## hp           -1.818e-02  1.371e-02  -1.326 0.185488
## w            -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acc           7.910e-02  9.822e-02   0.805 0.421101
## year          7.770e-01  5.178e-02  15.005  < 2e-16 ***
## factor(ori)2  2.630e+00  5.664e-01   4.643 4.72e-06 ***
## factor(ori)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
```

i. Which factors have a 'significant' effect on mpg at 1% significance?

`disp`, `weight`, `year`, and `origin`

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?

It is hard to determine which independent variables are the most effective at increasing mpg, because they are all have different units.

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

i. Create fully standardized regression results: are these values easier to interpret?

```r
auto_std <- data.frame(scale(auto[c(1:7)]))
auto_std$ori <- auto$ori
regr_mpg_std <- lm(mpg ~ cyl + disp + hp + w + acc + year + factor(ori),
                   data = auto_std)
summary(regr_mpg_std)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + w + acc + year + factor(ori),
##     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 ***
## cyl          -0.10658    0.06991  -1.524  0.12821
## disp          0.31989    0.10210   3.133  0.00186 **
## hp           -0.08955    0.06751  -1.326  0.18549
## w            -0.72705    0.07098 -10.243  < 2e-16 ***
## acc           0.02791    0.03465   0.805  0.42110
## year          0.36760    0.02450  15.005  < 2e-16 ***
## factor(ori)2  0.33649    0.07247   4.643 4.72e-06 ***
## factor(ori)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, it is easier to interpret.

ii. Regress mpg over each nonsignificant independent variable, individually. Which ones are significant if we regress mpg over them individually?

```r
sapply(auto_std[, c(2, 4, 6)], function(x) {
  summary(lm(auto_std$mpg ~ x))
})
```

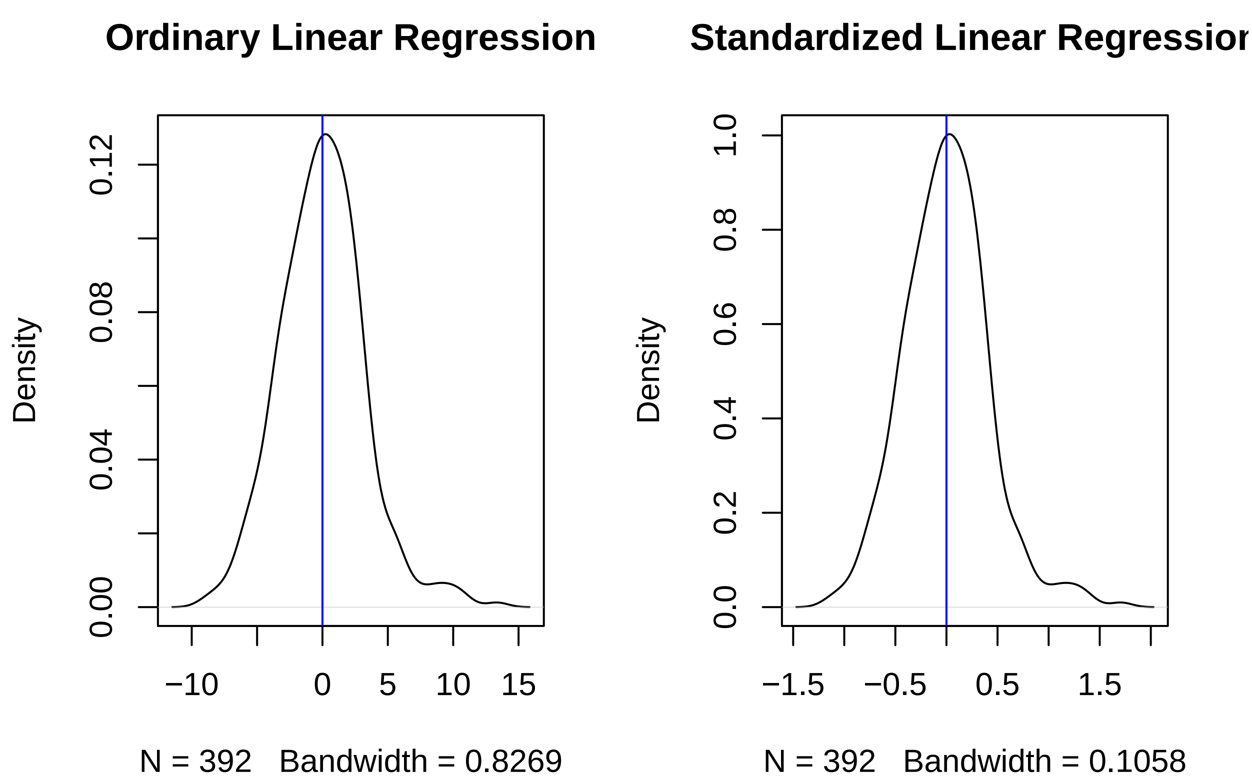
```
## $cyl
##
## Call:
## lm(formula = auto_std$mpg ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82455 -0.43297 -0.08288  0.32674  2.29046
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.834e-15  3.169e-02    0.00        1
## x           -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
##
##
## $hp
##
## Call:
## lm(formula = auto_std$mpg ~ x)
##
## 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
## x           -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
##
##
## $acc
##
## Call:
## lm(formula = auto_std$mpg ~ x)
##
## 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
## x           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
```

All of them (`cyl`, `hp`, and `acc`) are significant if we regress mpg over them individually.

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

```r
par(mfrow = c(1, 2))
plot(density(regr_mpg$residuals), main = 'Ordinary Linear Regression')
abline(v = 0, col = 'blue')
plot(density(regr_mpg_std$residuals), main = 'Standardized Linear Regression')
abline(v = 0, col = 'blue')
```



They both are centered around zero and almost normally distributed.