BACS HW12

Bijon Setyawan Raya

5/12/2021

Question 1

```
cars_log <- with(cars, data.frame(log(mpg), log(cylinders), log(displacement),
log(horsepower), log(weight), log(acceleration), year, origin))

names(cars_log) <- names(cars)[1:8] # rename the columns
head(cars_log)
```

```
##        mpg cylinders displacement horsepower   weight acceleration year origin
## 1 2.890372  2.079442     5.726848   4.867534 8.161660     2.484907   70      1
## 2 2.708050  2.079442     5.857933   5.105945 8.214194     2.442347   70      1
## 3 2.890372  2.079442     5.762051   5.010635 8.142063     2.397895   70      1
## 4 2.772589  2.079442     5.717028   5.010635 8.141190     2.484907   70      1
## 5 2.833213  2.079442     5.710427   4.941642 8.145840     2.351375   70      1
## 6 2.708050  2.079442     6.061457   5.288267 8.375860     2.302585   70      1
```

a. Run a new regression on the cars_log dataset, with mpg.log. dependent on all other variables

```
cars_log_regr <-
  lm(
    mpg ~
      cylinders +
      displacement +
      horsepower +
      weight +
      acceleration +
      year +
      factor(origin),
    data = cars_log
  )
summary(cars_log_regr)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39727 -0.06880  0.00450  0.06356  0.38542
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.301938   0.361777  20.184  < 2e-16 ***
## cylinders        -0.081915   0.061116  -1.340  0.18094
## displacement      0.020387   0.058369   0.349  0.72707
## horsepower       -0.284751   0.057945  -4.914 1.32e-06 ***
## weight           -0.592955   0.085165  -6.962 1.46e-11 ***
## acceleration     -0.169673   0.059649  -2.845  0.00469 **
## year              0.030239   0.001771  17.078  < 2e-16 ***
## factor(origin)2   0.050717   0.020920   2.424  0.01580 *
## factor(origin)3   0.047215   0.020622   2.290  0.02259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.113 on 383 degrees of freedom
## Multiple R-squared:  0.8919, Adjusted R-squared:  0.8897
## F-statistic:   395 on 8 and 383 DF,  p-value: < 2.2e-16
```

i. Which log-transformed factors have a significant effect on log.mpg. at 10% significance?

`horsepower`, `weight`, `acceleration`, `year`, `factor(origin)2`, and `factor(origin)3`.

ii. Do some new factors now have effects on mpg, and why might this be?

`acceleration` and `horsepower` suddenly became significant in this case, which they weren't in the previous homework.

iii. Which factors still have insignificant or opposite (from correlation) effects on mpg? Why might this be?

Only `cylinders`. The more cylinders cars have, the higher the gas consumption.

b. Let's take a closer look at weight, because it seems to be a major explanation of mpg

i. Create a regression (call it regr_wt) of mpg on weight from the original cars data set

```
regr_wt <- lm(mpg ~ weight, data = Auto)
```

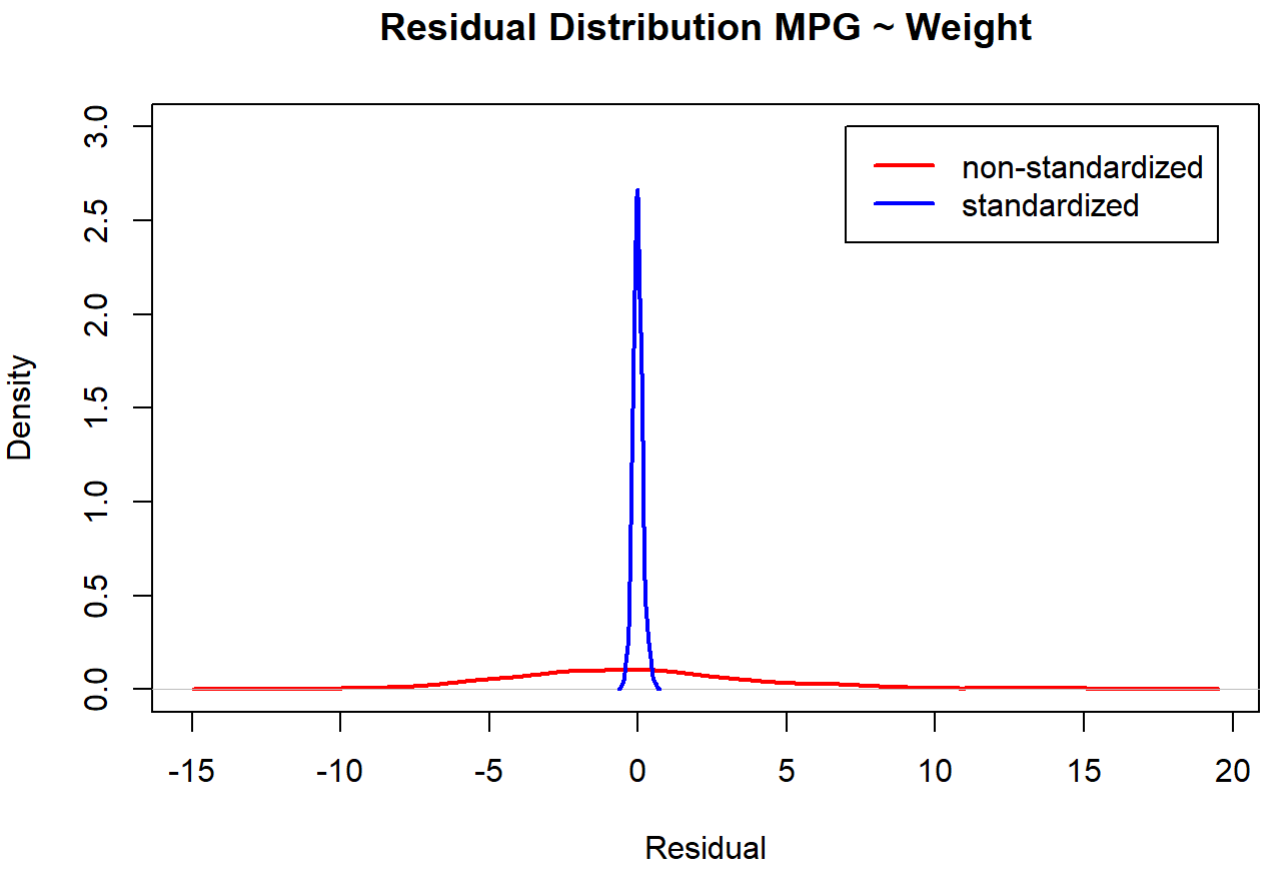
ii. Create a regression (call it regr_wt_log) of log.mpg. on log.weight. from cars_log

```
regr_wt_log <- lm(mpg ~ weight, data = cars_log)
```

iii. visualize the residuals of both regression models

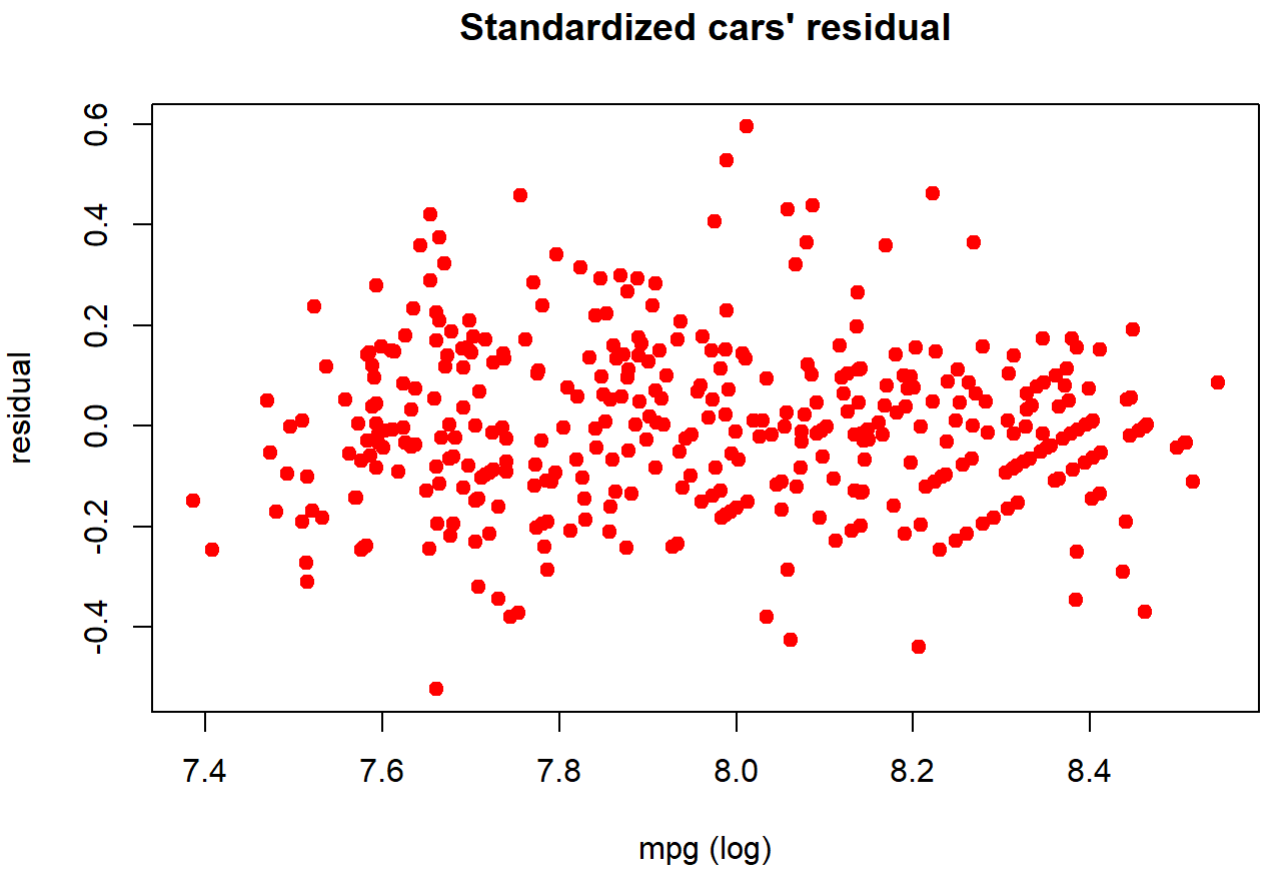
1. density plots of residuals

```
plot(
  density(resid(regr_wt)),
  main = "Residual Distribution MPG ~ Weight",
  lwd = 2,
  col = "red",
  xlab = "Residual",
  ylim = c(0, 3)
)
lines(density(resid(regr_wt_log)), lwd = 2, col = "blue")
legend(
  7,
  3,
  c("non-standardized", "standardized"),
  col = c("red", "blue"),
  lwd = c(2,2),
  lty = c("solid", "solid")
)
```



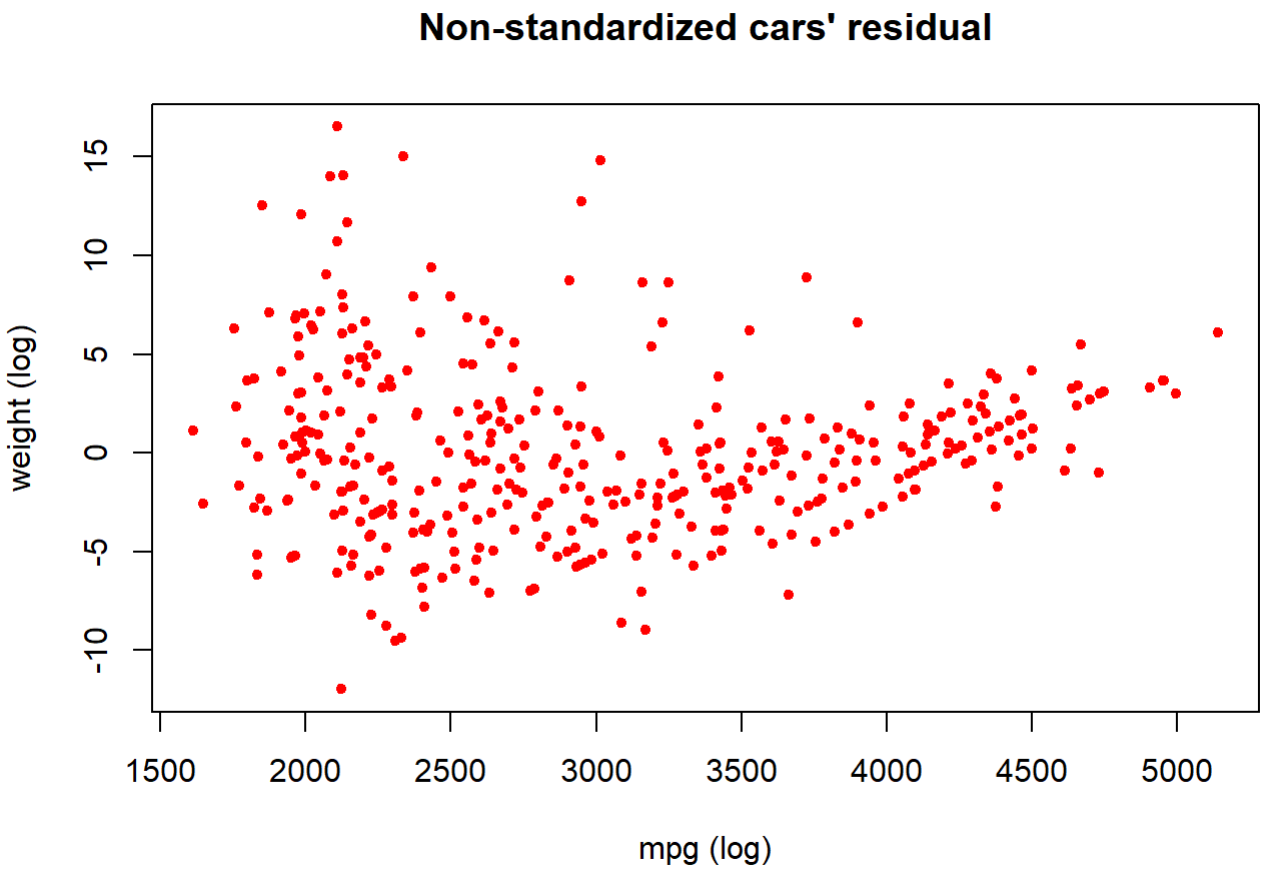
2. scatterplot of log.weight. vs. residuals

```
plot(
  cars_log$weight,
  regr_wt_log$residuals,
  col = "red",
  pch = 19,
  xlab = "mpg (log)",
  ylab = "residual",
  main = "Standardized cars' residual"
)
```



iv. Which regression produces better residuals for the assumptions of regression?

```
plot(
  Auto$weight,
  regr_wt$residuals,
  col = "red",
  pch = 20,
  xlab = "mpg (log)",
  ylab = "weight (log)",
  main = "Non-standardized cars' residual"
)
```

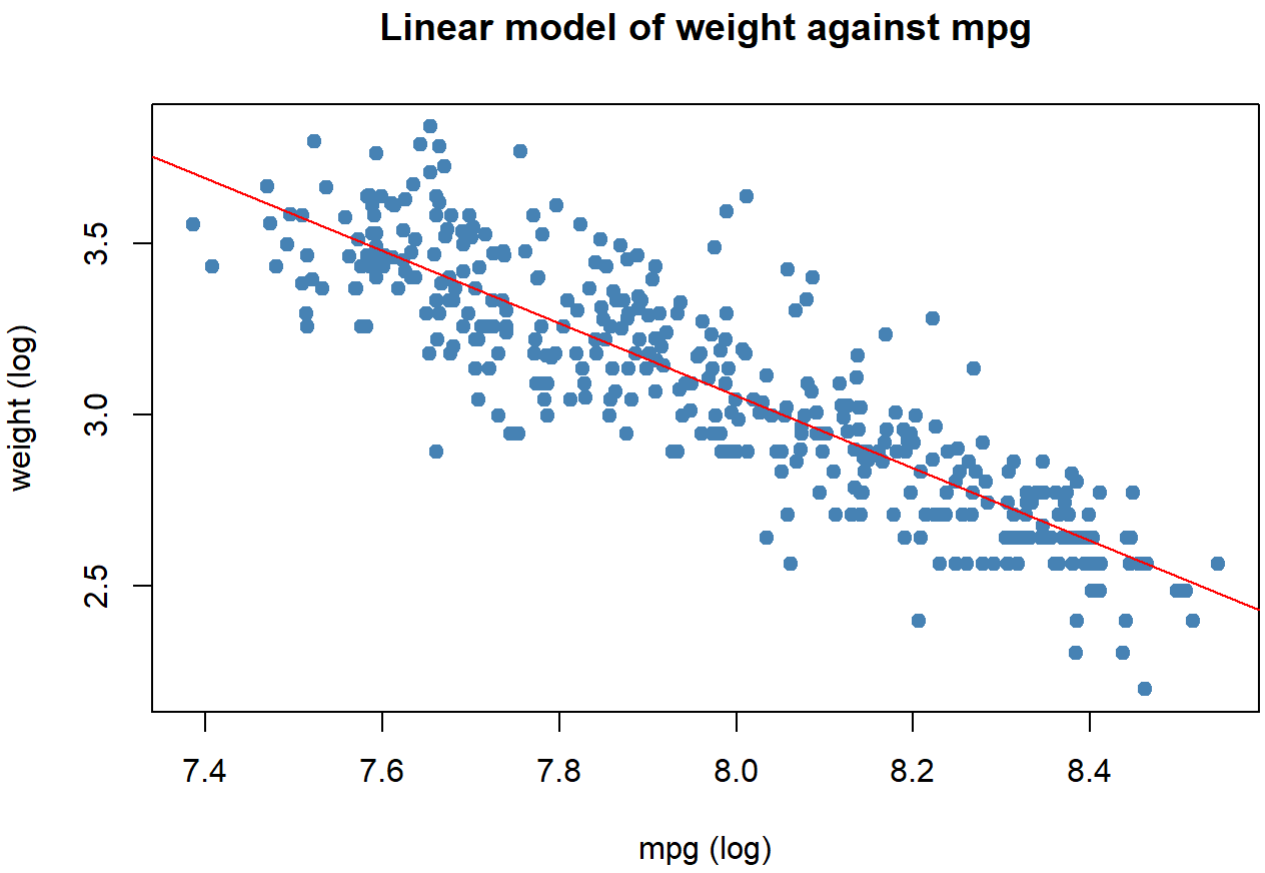


Looking at this graph and the previous graph (the standardized one), we can tell that most data points are centralized in the middle. Thus, the standardized one produces better residuals.

v. How would you interpret the slope of log.weight. vs log.mpg. in simple words?

```
plot(
  cars_log$weight,
  cars_log$mpg,
  col = "steelblue",
  pch = 19,
  xlab = "mpg (log)",
  ylab = "weight (log)",
  main = "Linear model of weight against mpg"
)

abline(
  a = regr_wt_log$coefficients["(Intercept)"],
  b = regr_wt_log$coefficients["weight"],
  col = "red"
)
```



Clearly, the lighter the cars, the further the distance can be covered per gallon.

```
lm(cars_log$mpg ~ cars_log$weight)
```

```
##
## Call:
## lm(formula = cars_log$mpg ~ cars_log$weight)
##
## Coefficients:
##     (Intercept)  cars_log$weight
##          11.515           -1.058
```

The summary above also means that 1% change in mpg leads to 1% decrease in weight.

c. Let's examine the 95% confidence interval of the slope of log.weight. vs. log.mpg.

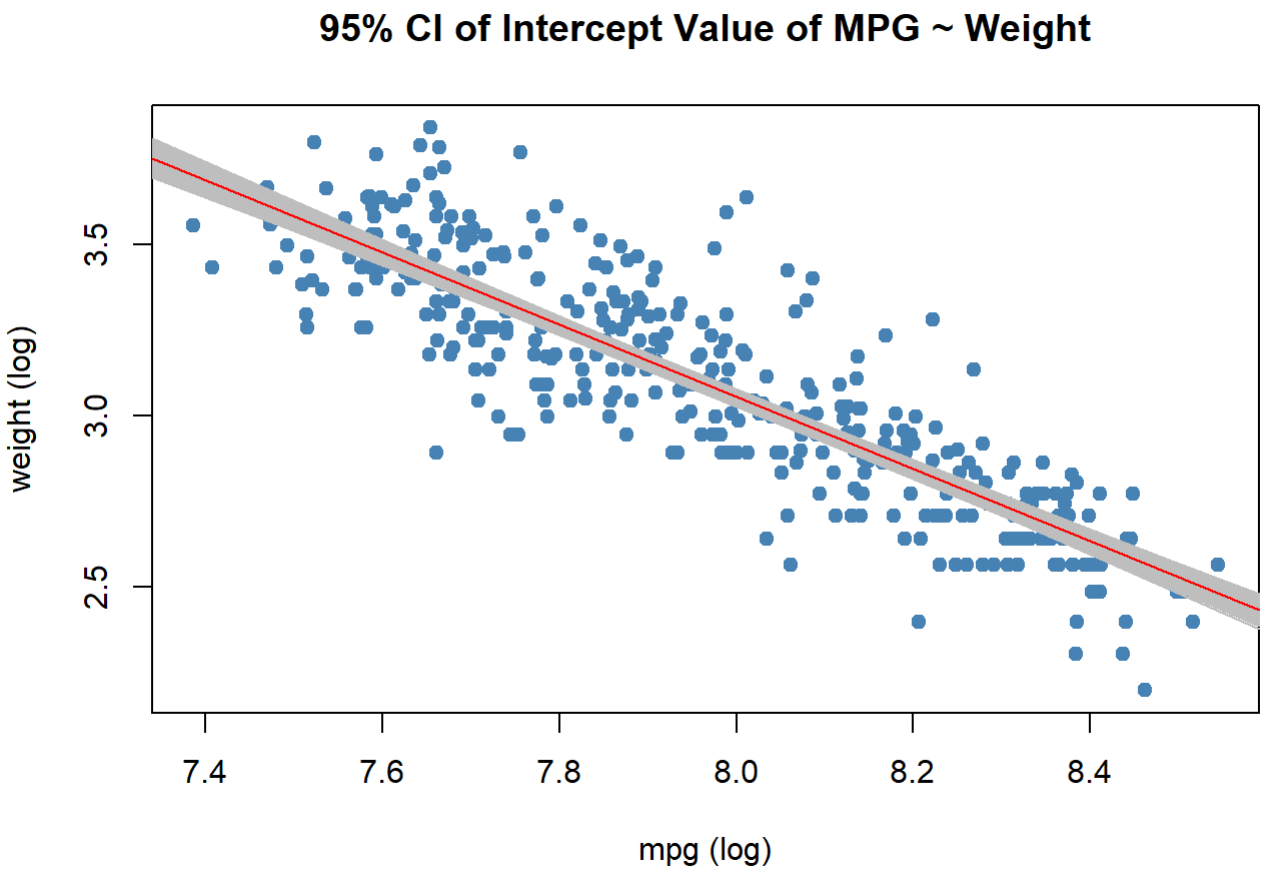
i. Create a bootstrapped confidence interval

```r
boot_intercept <- function(dataset) {
  # get random data points' indexes
  indexes <- sample(1:nrow(dataset), replace = TRUE)
  slopes <- lm(mpg ~ weight, data = dataset[indexes,])
  abline(slopes, lwd = 1, col="grey")
  return(slopes$coefficients)
}
```

```r
plot(
  cars_log$weight,
  cars_log$mpg,
  col = "steelblue",
  pch = 19,
  xlab = "mpg (log)",
  ylab = "weight (log)",
  main = "95% CI of Intercept Value of MPG ~ Weight"
)

regression_coeffs <- replicate(500, boot_intercept(cars_log))

abline(
  a = mean(regression_coeffs["(Intercept)",]),
  b = mean(regression_coeffs["weight",]),
  col = "red"
)
```



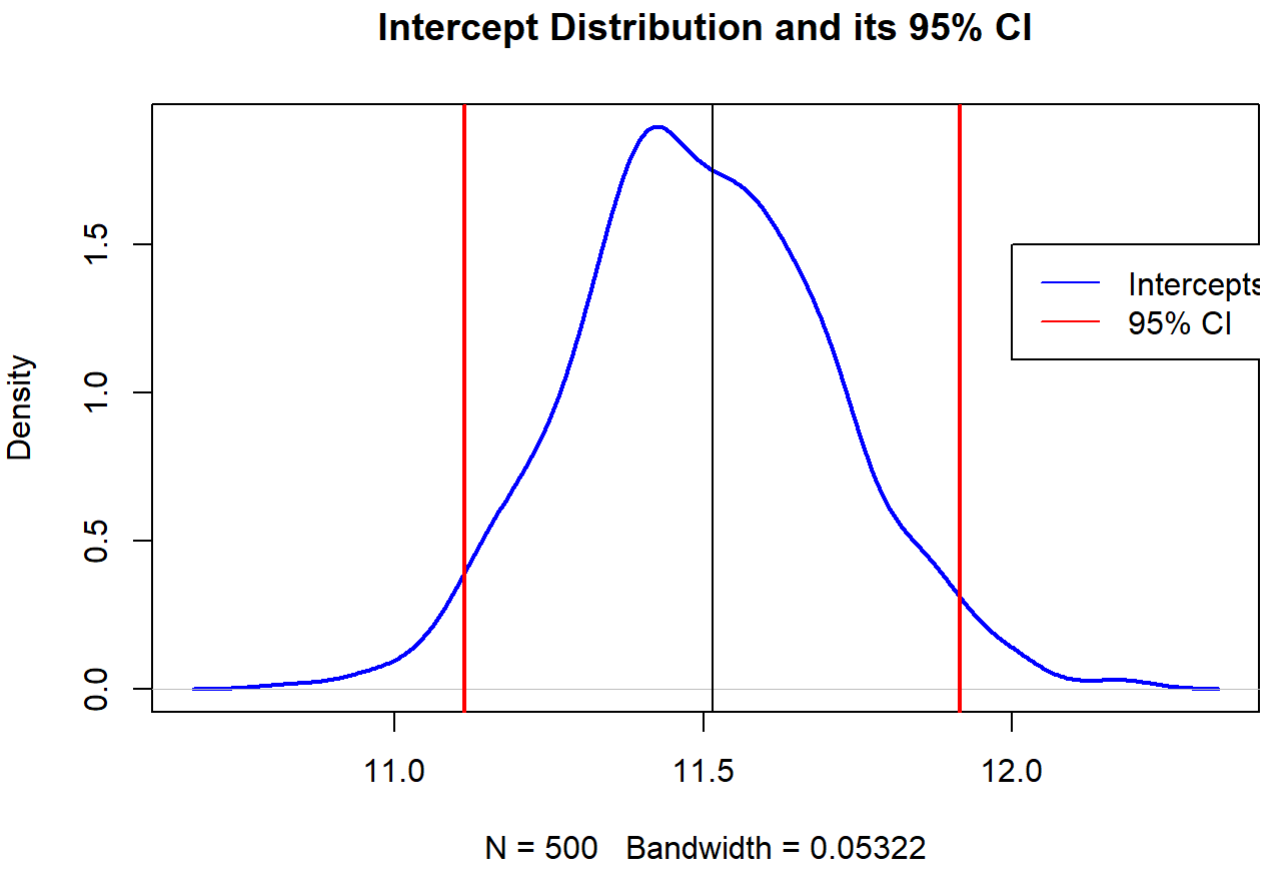
ii. Verify your results with a confidence interval using traditional statistics

```
plot(
  density(regression_coeffs["(Intercept)",]),
  lwd = 2,
  col="blue",
  main = "Intercept Distribution and its 95% CI"
)

abline(
  v = lm(formula = mpg ~ weight, data = cars_log)$coefficients['(Intercept)']
)

abline(
  v = quantile(
      regression_coeffs["(Intercept)",],
      probs = c(0.025, 0.975)
    ),
  col = "red",
  lwd = 2
)

legend(
  12,
  1.5,
  c("Intercepts", "95% CI"),
  col = c('blue', 'red'),
  lty = c("solid", "solid")
)
```



We can see that the regression intercept from `lm(formula = mpg ~ weight, data = cars_log)` falls in the 95% CI.

Question 2

```
regr_log <-
  lm(
    mpg ~
      cylinders +
      displacement +
      horsepower +
      weight +
      acceleration +
      year +
      factor(origin),
    data = cars_log
  )
```

a. Using regression and R2, compute the VIF of log.weight. using the approach shown in class

```
weight_regr <-
  lm(weight ~ cylinders + displacement + horsepower
            + acceleration + year + factor(origin),
     data=cars_log,
     na.action = na.exclude)

r2_weight <- summary(weight_regr)$r.squared
vif <- 1 / (1-r2_weight)
vif
```

```
## [1] 17.57512
```

The result above means that `weight` shares more than half of its variance with other independent variables.

b. Let's try a procedure called Stepwise VIF Selection to remove highly collinear predictors.

i. Use vif(regr_log) to compute VIF of the all the independent variables

```
regr_log_vif <- vif(regr_log)
regr_log_vif
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## cylinders      10.456738  1        3.233688
## displacement   29.625732  1        5.442952
## horsepower     12.132057  1        3.483110
## weight         17.575117  1        4.192269
## acceleration    3.570357  1        1.889539
## year            1.303738  1        1.141814
## factor(origin)  2.656795  2        1.276702
```

ii. Eliminate from your model the single independent variable with the largest VIF score that is also greater than 5

```
regr_log <-
  lm(
    mpg ~
      cylinders +
      horsepower +
      weight +
      acceleration +
      year +
      factor(origin),
    data = cars_log
  )
regr_log_vif <- vif(regr_log)
regr_log_vif
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## cylinders       5.433107  1        2.330903
## horsepower     12.114475  1        3.480585
## weight         11.239741  1        3.352572
## acceleration    3.327967  1        1.824272
## year            1.291741  1        1.136548
## factor(origin)  1.897608  2        1.173685
```

iii. Repeat steps (i) and (ii) until no more independent variables have VIF scores above 5

```
regr_log <-
  lm(
    mpg ~
      cylinders +
      weight +
      acceleration +
      year +
      factor(origin),
    data = cars_log
  )
regr_log_vif <- vif(regr_log)
regr_log_vif
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## cylinders      5.427610  1        2.329723
## weight         4.871730  1        2.207200
## acceleration   1.401202  1        1.183724
## year           1.206351  1        1.098340
## factor(origin) 1.821167  2        1.161682
```

In this iteration, we have to remove `cylinders` which has the highest VIF.

```
regr_log <-
  lm(
    mpg ~
      weight +
      acceleration +
      year +
      factor(origin),
    data = cars_log
  )
regr_log_vif <- vif(regr_log)
regr_log_vif
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## weight          1.933208  1        1.390398
## acceleration    1.304761  1        1.142261
## year            1.175545  1        1.084225
## factor(origin) 1.710178  2        1.143564
```

At the end, only `weight`, `acceleration`, `year`, and `factor(origin)`.

iv. Report the final regression model and its summary statistics

```
regr_log
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + year + factor(origin),
##     data = cars_log)
##
## Coefficients:
##     (Intercept)           weight     acceleration             year
##         7.41097         -0.87550          0.05438          0.03279
## factor(origin)2  factor(origin)3
##         0.05611          0.03194
```

```
summary(regr_log)
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + year + factor(origin),
##     data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38259 -0.07054  0.00401  0.06696  0.39798
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.410974   0.316806  23.393  < 2e-16 ***
## weight           -0.875499   0.029086 -30.101  < 2e-16 ***
## acceleration      0.054377   0.037132   1.464  0.14389
## year              0.032787   0.001731  18.937  < 2e-16 ***
## factor(origin)2   0.056111   0.018241   3.076  0.00225 **
## factor(origin)3   0.031937   0.018506   1.726  0.08519 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1163 on 386 degrees of freedom
## Multiple R-squared:  0.8845, Adjusted R-squared:  0.883
## F-statistic: 591.1 on 5 and 386 DF,  p-value: < 2.2e-16
```

c. Using stepwise VIF selection, have we lost any variables that were previously significant?

Without using step wise VIF selection, we have `horsepower`, `weight`, `acceleration`, `year`, `factor(origin)2`, and `factor(origin)3` as the significant independent variables.

However, we lose `horsepower`, `acceleration`, and `factor(origin)3` are lost using step wise VIF selection.

d. From only the formula for VIF, try deducing/deriving the following:

i. If an independent variable has no correlation with other independent variables, what would its VIF score be?

If VIF is between 1 and 5, then the variables moderately correlated. If the VIF is greater than 5, then the variables are highly correlated.

If VIF score would be less than one 1, then the variables are less correlated.

ii. Given a regression with only two independent variables (X1 and X2), how correlated would X1 and X2 have to be, to get VIF scores of 5 or higher? To get VIF scores of 10 or higher?

If the VIF score of 5 or higher, the correlation will be

```
vif <- 5
correlation <- sqrt(1-(1/vif))
correlation
```

```
## [1] 0.8944272
```

If the VIF score of 10 or higher, the correlation will be

```
vif <- 10
correlation <- sqrt(1-(1/vif))
correlation
```

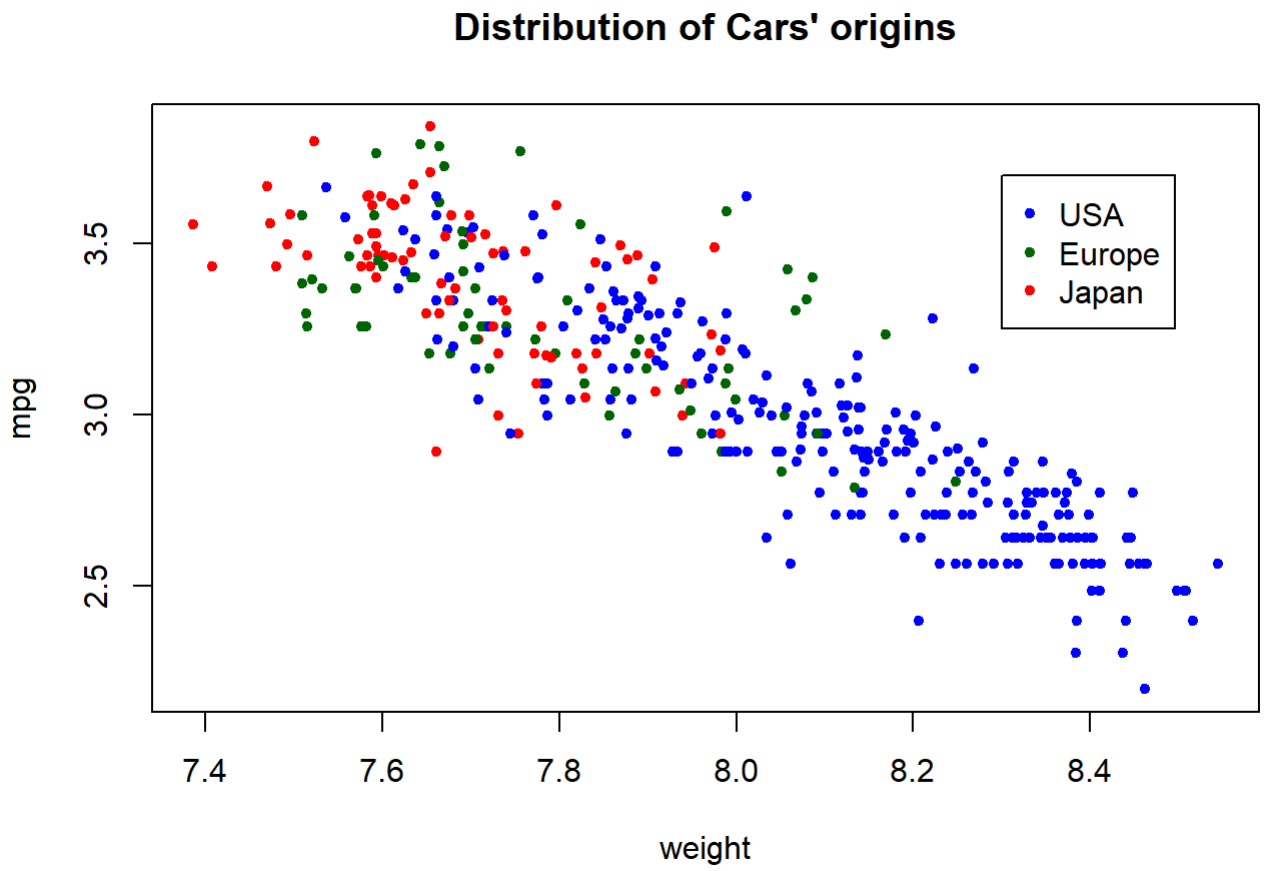
```
## [1] 0.9486833
```

Question 3

```
origin_colors = c("blue", "darkgreen", "red")

with(
  cars_log,
  plot(
    weight,
    mpg,
    pch=20,
    col=origin_colors[origin],
    main = "Distribution of Cars' origins"
  )
)

legend(
  8.3,
  3.7,
  c("USA", "Europe", "Japan"),
  col = c("blue", "darkgreen", "red"),
  pch = c(20,20,20),
)
```



a. Let's add three separate regression lines on the scatterplot, one for each of the origins:

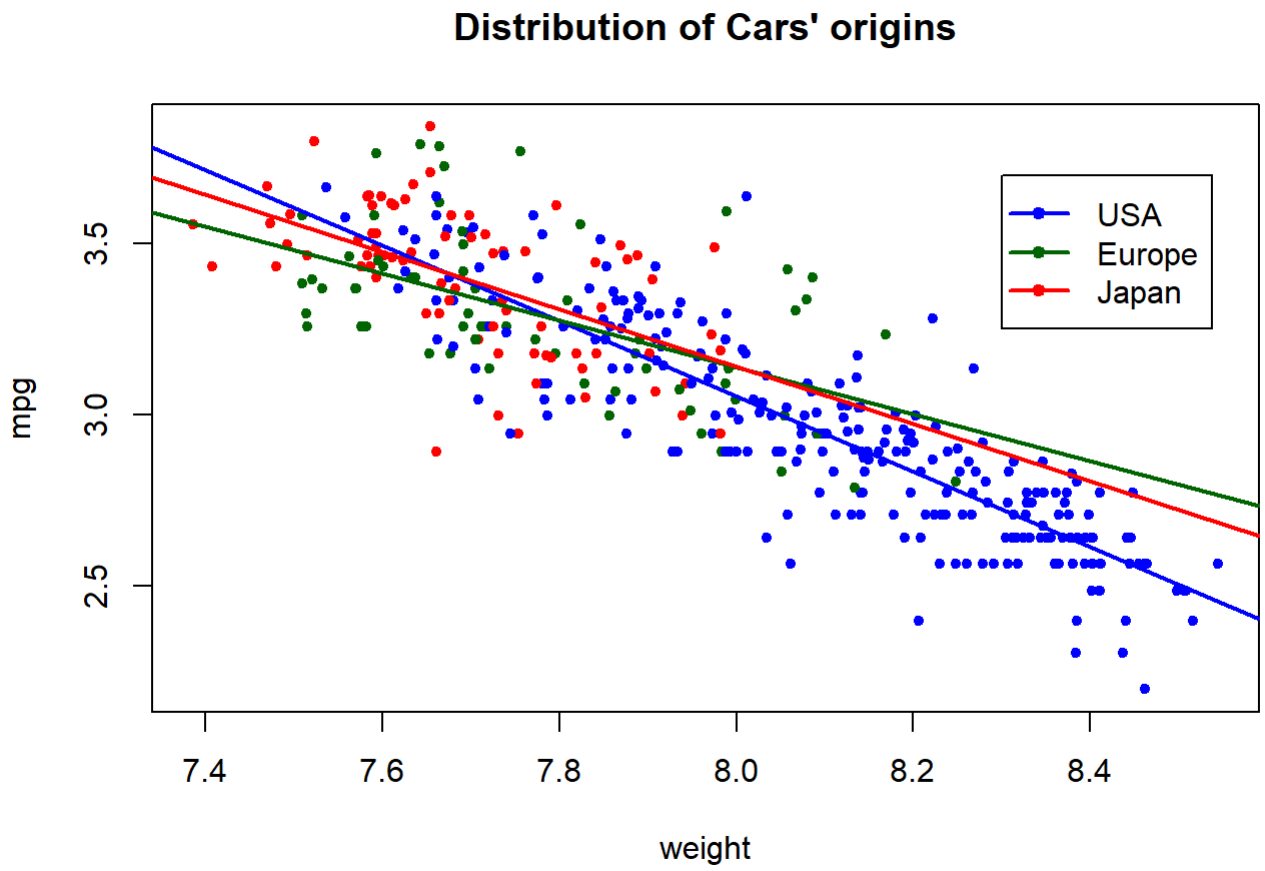
```
origin_colors = c("blue", "darkgreen", "red")
with(
  cars_log,
  plot(
    weight,
    mpg,
    pch=20,
    col=origin_colors[origin],
    main = "Distribution of Cars' origins"
  )
)

abline(
  lm(
    mpg~weight,
    data=cars_log[cars_log$origin == 1,]
  ),
  col = origin_colors[1],
  lwd = 2
)

abline(
  lm(
    mpg~weight,
    data=cars_log[cars_log$origin == 2,]
  ),
  col=origin_colors[2],
  lwd=2
)

abline(
  lm(
    mpg~weight,
    data=cars_log[cars_log$origin == 3,]
  ),
  col=origin_colors[3],
  lwd=2
)

legend(
  8.3,
  3.7,
  c("USA", "Europe", "Japan"),
  col = c("blue", "darkgreen", "red"),
  lty = c(1,1,1),
  lwd = c(2,2,2),
  pch = c(20,20,20)
)
```



b. Do cars from different origins appear to have different weight vs. mpg relationships?

It doesn’t seem like it. Cars from those countries show the same trend where the ligher the cars, the further the distance the cars can cover per gallon.