BACS HW (Week 11)

108020024

due on 04/30 (Sun)

Question 1) Let's deal with nonlinearity first. Create a new dataset that log-transforms several variables from our original dataset (called cars in this case):

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

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

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

```
md1 <- lm(log.mpg. ~ . -origin +factor(origin), , data = cars_log)
summary(md1)
```

```
##
## Call:
## lm(formula = log.mpg. ~ . - origin + 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 ***
## log.cylinders.     -0.081915   0.061116  -1.340  0.18094
## log.displacement.   0.020387   0.058369   0.349  0.72707
## log.horsepower.    -0.284751   0.057945  -4.914 1.32e-06 ***
## log.weight.        -0.592955   0.085165  -6.962 1.46e-11 ***
## log.acceleration. -0.169673   0.059649  -2.845  0.00469 **
## model_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
##   (    6      )
## 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?

log.horsepower. log.weight. log.acceleration.

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

Compare with the model last week, yes there are some new factors that are significant on log.mpg. The new factors are:

log.horsepower. log.acceleration.

Because in the last homework, from the scatter plot of "mpg and horsepower", "mpg and acceleration", we can find non-linear relationships between them, and this will cause problem while doing linear regression. There are different way to handle non-linear relationship, (it depends from the diagnosing), **taking log transform one or both sides of our regression is a quick way to fix the non-linear relationships in our model for this data set.**

For more information, check NTHU STAT 5410 - Linear Models:

http://www.stat.nthu.edu.tw/~swcheng/Teaching/stat5410/index.php?fbclid=IwAR1pUzJe_tmLx0wyOBxFZqHCk8jIB1E UDvCktVD86uRP77TUxUclN_NY

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

The variables that are insignificant on mpg are: **log.cylinders. log.displacement.** The reason might cause by the high multicollinearity against cylinders, displacement, horsepower,and weight.

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 over weight from the original cars dataset

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

```
##
## Call:
## lm(formula = mpg ~ weight, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.012  -2.801  -0.351   2.114  16.480
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.3173644  0.7952452   58.24   <2e-16 ***
```

```
## weight      -0.0076766  0.0002575  -29.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.345 on 396 degrees of freedom
## Multiple R-squared:  0.6918, Adjusted R-squared:  0.691
## F-statistic: 888.9 on 1 and 396 DF,  p-value: < 2.2e-16
```

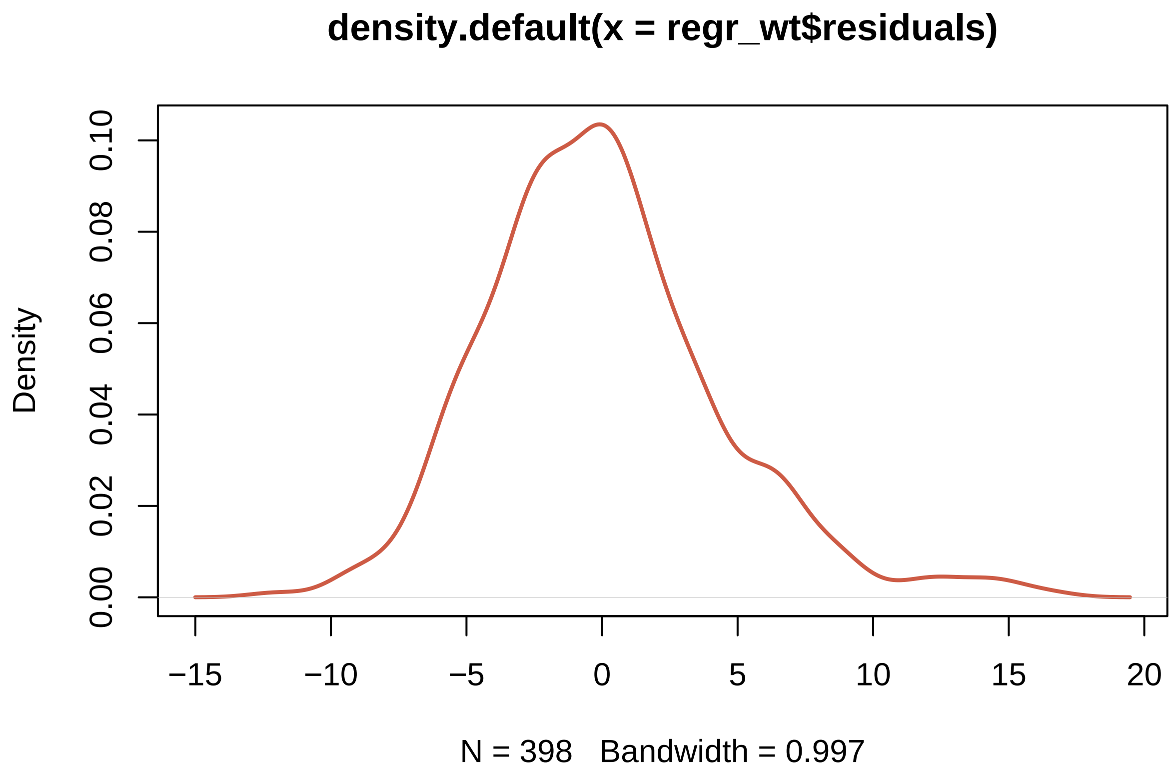
ii) Create a regression (call it regr_wt_log) of log.mpg. on log.weight. from cars_log

```r
regr_wt_log <- lm(log.mpg. ~log.weight., data = cars_log)
summary(regr_wt_log)
```

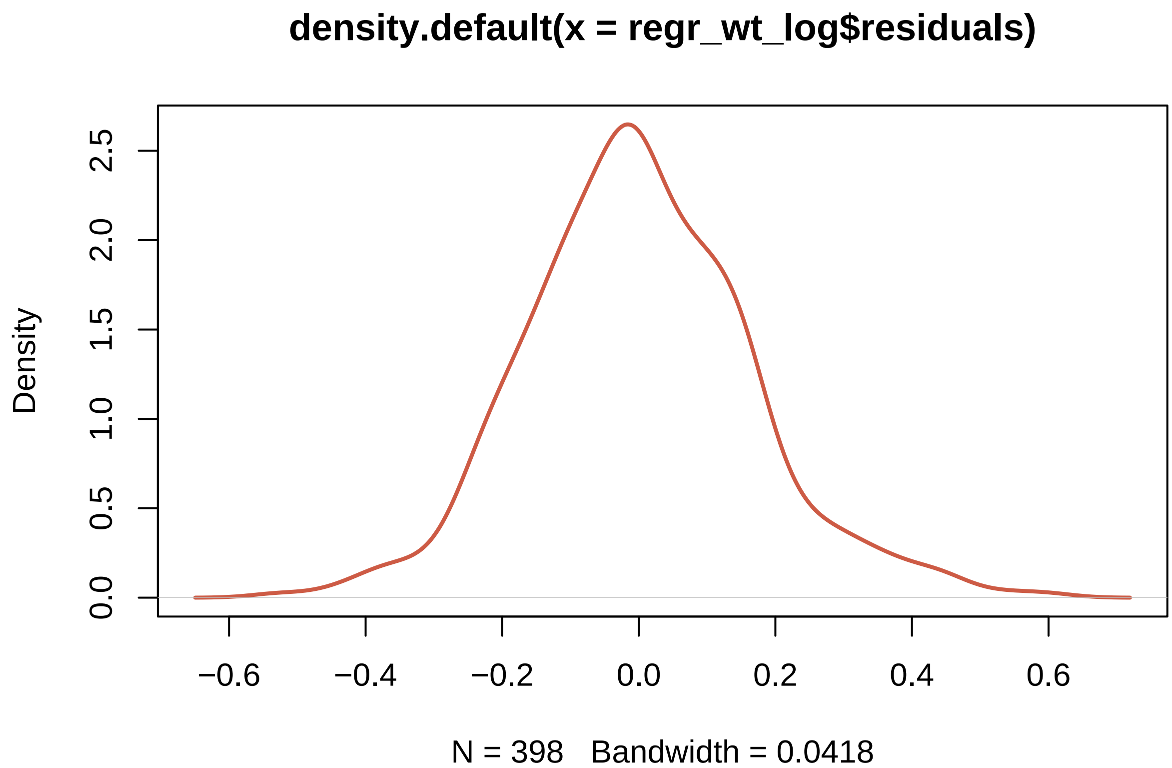
```
##
## Call:
## lm(formula = log.mpg. ~ log.weight., data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52408 -0.10441 -0.00805  0.10165  0.59384
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.5219     0.2349   49.06   <2e-16 ***
## log.weight.  -1.0583     0.0295  -35.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.165 on 396 degrees of freedom
## Multiple R-squared:  0.7647, Adjusted R-squared:  0.7641
## F-statistic:  1287 on 1 and 396 DF,  p-value: < 2.2e-16
```

iii) Visualize the residuals of both regression models (raw and log-transformed):
  1.density plots of residuals

```r
plot(density(regr_wt$residuals), col="coral3", lwd=2)
```




```
plot(density(regr_wt_log$residuals), col="coral3", lwd=2)
```

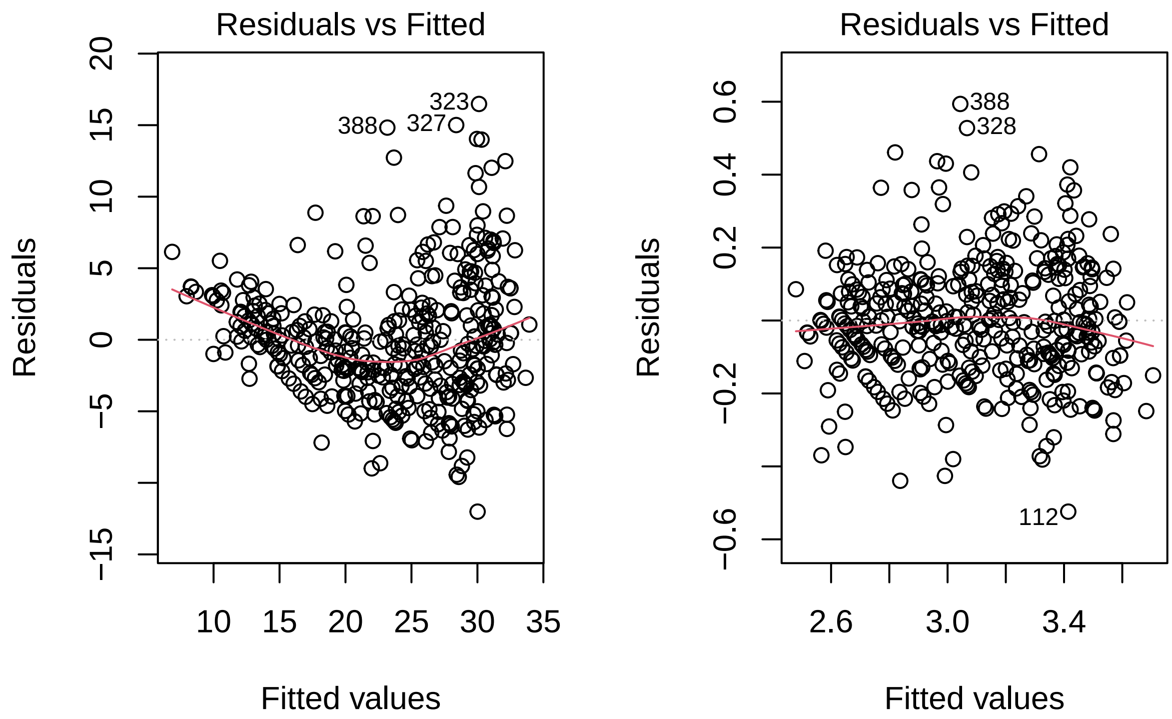


2.scatterplot of log.weight. vs. residuals

‘

```
par(mfrow=c(1,2))

plot(regr_wt, which = c(1,1))
plot(regr_wt_log, which = c(1,1))
```




```r
par(mfrow=c(1,1))
```

```
iv) Which regression produces better distributed residuals for the assumptions of regression?
```

regr_wt_log produces better distributed residuals for the assumptions of regression.

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

The slope of log.weight. vs log.mpg. is nearly horizontal.

```
vi)From its standard error, what is the 95% confidence interval of the slope of log.weight. vs log.
```

```r
#95% CI
c(-1.0583-1.96*0.0295,-1.0583+1.96*0.0295)
```

```
## [1] -1.11612 -1.00048
```

The 95% confidence interval of the slope of log.weight. vs log.mpg. is (-1.11612 -1.00048)

Question 2) Let's tackle multicollinearity next.

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

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

```r
regr_log.weight. <- lm(log.weight. ~ log.cylinders. + log.displacement. + log.horsepower.
                               + log.acceleration. + model_year +
                              factor(origin), data=cars_log)


r2_log.weight. <- summary(regr_log.weight.)$r.squared
vif_log.weight. <- 1 / (1 - r2_log.weight.)
vif_log.weight.
```

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

The VIF of log.weight. is 9.251547.

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
```

```r
library(car)
```

```
##     carData
```

```r
vif(regr_log)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## log.cylinders.    10.456738  1        3.233688
## log.displacement. 29.625732  1        5.442952
## log.horsepower.   12.132057  1        3.483110
## log.weight.       17.575117  1        4.192269
## log.acceleration.  3.570357  1        1.889539
## model_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

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

eliminate log.displacement.

```r
regr_log <- lm(log.mpg. ~ log.cylinders.  + log.horsepower. +
                              log.weight. + log.acceleration. + model_year +
                              factor(origin), data=cars_log)

vif(regr_log)
```

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

eliminate log.horsepower.

```r
regr_log <- lm(log.mpg. ~ log.cylinders. +
                              log.weight. + log.acceleration. + model_year +
                              factor(origin), data=cars_log)

vif(regr_log)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## log.cylinders.    5.321090  1        2.306749
## log.weight.       4.788498  1        2.188264
## log.acceleration. 1.400111  1        1.183263
## model_year        1.201815  1        1.096273
## factor(origin)    1.792784  2        1.157130
```

eliminate log.cylinders.

```r
regr_log <- lm(log.mpg. ~log.weight. + log.acceleration. + model_year +
                              factor(origin), data=cars_log)

vif(regr_log)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## log.weight.       1.926377  1        1.387940
## log.acceleration. 1.303005  1        1.141493
## model_year        1.167241  1        1.080389
## factor(origin)    1.692320  2        1.140567
```

```
    iv) Report the final regression model and its summary statistics
```

The final model is lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year + factor(origin), data = cars_log).

```r
summary(regr_log)
```

```
##
## Call:
## lm(formula = log.mpg. ~ log.weight. + log.acceleration. + model_year +
##     factor(origin), data = cars_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38275 -0.07032  0.00491  0.06470  0.39913
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.431155   0.312248  23.799  < 2e-16 ***
## log.weight.       -0.876608   0.028697 -30.547  < 2e-16 ***
## log.acceleration.  0.051508   0.036652   1.405  0.16072
## model_year         0.032734   0.001696  19.306  < 2e-16 ***
## factor(origin)2    0.057991   0.017885   3.242  0.00129 **
## factor(origin)3    0.032333   0.018279   1.769  0.07770 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1156 on 392 degrees of freedom
## Multiple R-squared:  0.8856, Adjusted R-squared:  0.8841
## F-statistic: 606.8 on 5 and 392 DF,  p-value: < 2.2e-16
```

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

If so, how much did we hurt our explanation by dropping those variables? (hint: look at model fit)

We lost log.horsepower. which was previously significant.

The R^2 goes from 0.8919 to 0.8856, 0.8919 - 0.8856 = 0.0063, so we only loss about 0.0063 of the explanation of variation for the model.

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?
```

Its VIF score should be 1, because it has no correlation with other independent variables, it's r^2 is 0, and by the formula. 1/(1-0) = 1.

```
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?
```

To get VIF scores of 5 or higher, r^2 for that variable is at least 0.8, so the corrlation x1 and x2 will be sqrt(0.8), will be at least 0.8944272.

To get VIF scores of 10 or higher, r^2 for that variable is at least 0.9, so the corrlation x1 and x2 will be sqrt(0.9), will be at least 0.9486833.

Question 3) Might the relationship of weight on mpg be different for cars from different origins?

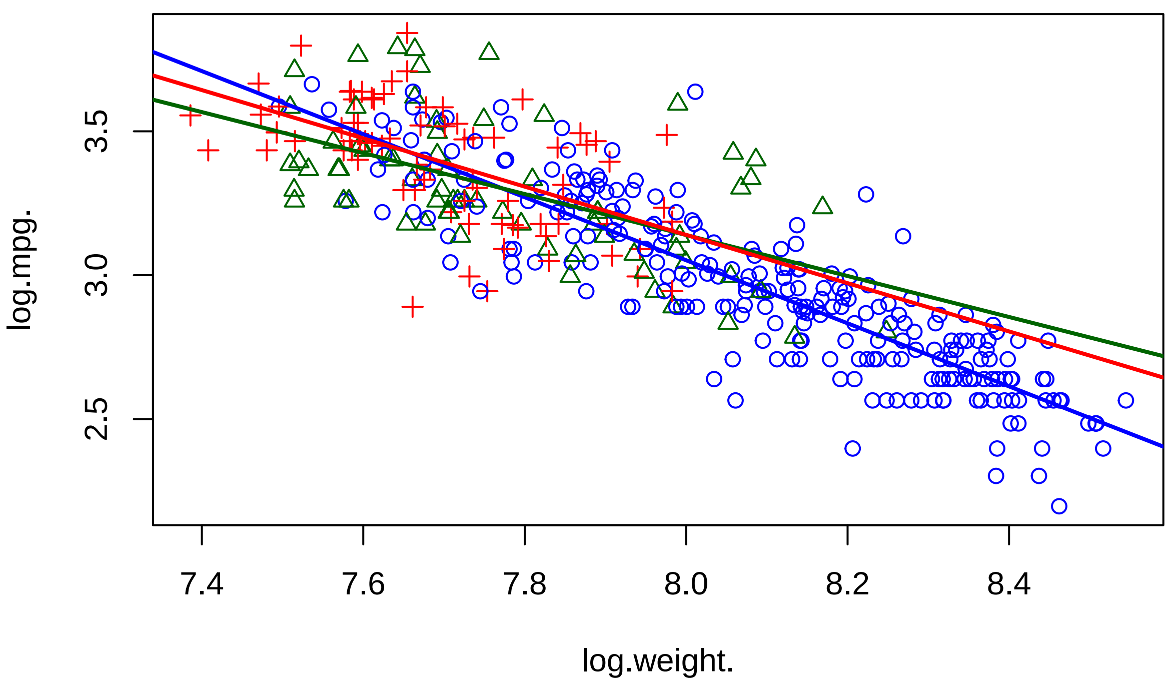
a) Let's add three separate regression lines on the scatterplot, one for each of the origins.

```
origin_colors = c("blue", "darkgreen", "red")
with(cars_log, plot(log.weight., log.mpg., pch=origin, col=origin_colors[origin]))

cars_us <- subset(cars_log, origin==1)
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_us)
abline(wt_regr_us, col=origin_colors[1], lwd=2)

cars_eu <- subset(cars_log, origin==2)
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_eu)
abline(wt_regr_us, col=origin_colors[2], lwd=2)

cars_jp <- subset(cars_log, origin==3)
wt_regr_us <- lm(log.mpg. ~ log.weight., data=cars_jp)
abline(wt_regr_us, col=origin_colors[3], lwd=2)
```



b)[not graded] Do cars from different origins appear to have different weight vs. mpg relationships?

There may need further modeling such as logistic regression to really know how origins affect the weight vs. mpg relationship. From the plot only, I guess that the blue dots, which represent US, may have heavier cars, since the blue dots are seperated from EU and JP, and most of them are in the right bottom part, which means heavy.