BACS-hw10-107070004

Question 1) Download demo_simple_regression_rsq.R from Canvas – it has a function that runs a regression simulation. This week, the simulation also reports R2 along with the other metrics from last week.

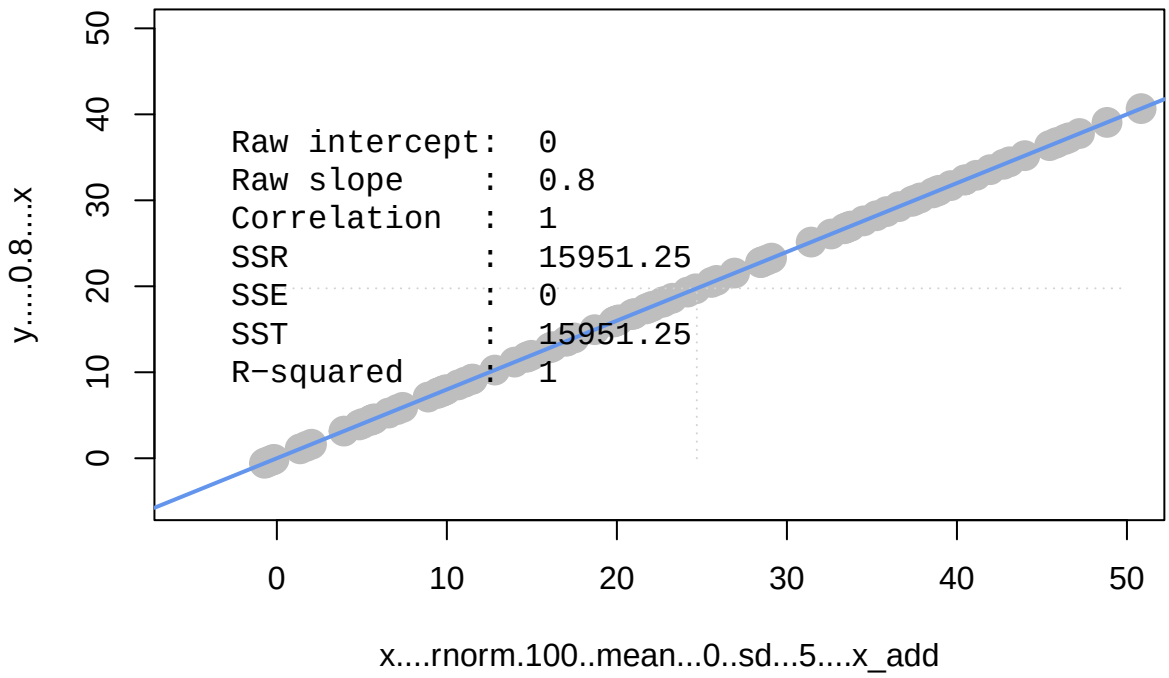
To answer the questions below, understand each of these four scenarios by simulating them: Scenario 1: Consider a very **narrowly dispersed** set of points that have a negative or positive **steep** slope Scenario 2: Consider a **widely dispersed** set of points that have a negative or positive **steep** 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

```
source("demo_simple_regression_rsq.R")
```

- Scenario 1 :

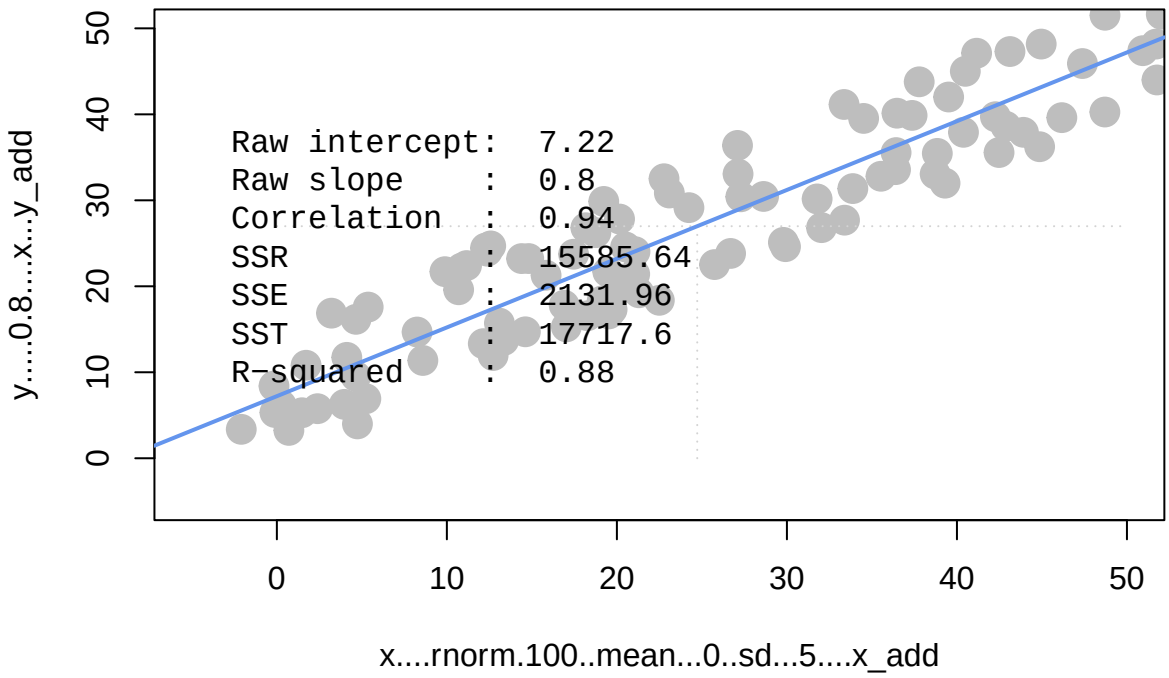
```
set.seed(2)
x_add = runif(100, 0, 50)
points_1 = data.frame(
  x <- rnorm(100, mean = 0, sd=5) + x_add,
  y <- (0.8) * x
)
plot_regr_rsq(points_1)
```

```
## Warning in summary.lm(regr): essentially perfect fit: summary may be unreliable
```



- Scenario 2 :

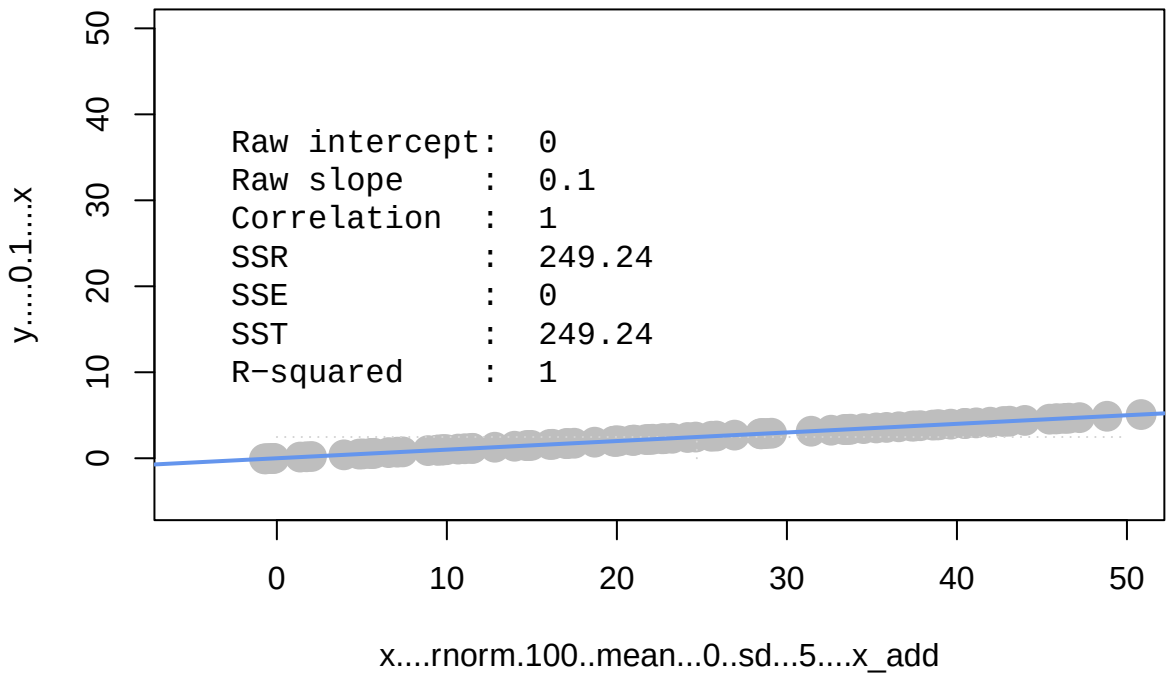
```
set.seed(2)
x_add = runif(100, 0, 50)
y_add = runif(100, 0, 15)
points_2 = data.frame(
  x <- rnorm(100, mean = 0, sd=5) + x_add,
  y <- (0.8) * x + y_add
)
plot_regr_rsq(points_2)
```



- Scenario 3 :

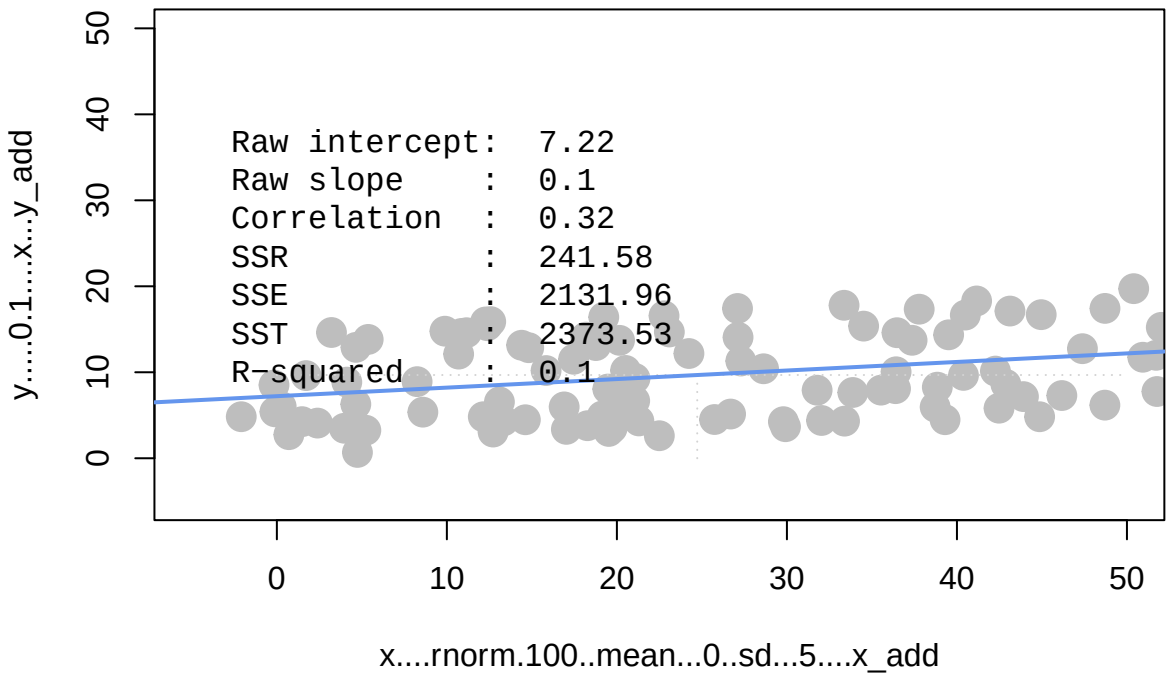
```
set.seed(2)
x_add = runif(100, 0, 50)
points_3 = data.frame(
  x <- rnorm(100, mean = 0, sd=5) + x_add,
  y <- (0.1) * x
)
plot_regr_rsq(points_3)
```

```
## Warning in summary.lm(regr): essentially perfect fit: summary may be unreliable
```



- Scenario 4

```
set.seed(2)
x_add = runif(100, 0, 50)
y_add = runif(100, 0, 15)
points_4 = data.frame(
  x <- rnorm(100, mean = 0, sd=5) + x_add,
  y <- (0.1) * x + y_add
)
plot_regr_rsq(points_4)
```



(a) Comparing scenarios 1 and 2, which do we expect to have a stronger R^2 ?

ans: scenarios 1 > scenarios 2

(b) Comparing scenarios 3 and 4, which do we expect to have a stronger R^2 ?

ans: scenarios 3 > scenarios 4

(c) Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

	scenarios 1	scenarios 2
SSE	smaller	bigger
SSR	bigger	smaller
SST	bigger	smaller

(d) Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

	scenarios 3	scenarios 4
SSE	smaller	bigger
SSR	bigger	smaller
SST	smaller	bigger

Question 2) Let's perform regression ourselves on the programmer_salaries.txt dataset we saw in class.

You can read the file using read.csv("programmer_salaries.txt", sep="")

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

(a) First, use the lm() function to estimate the model Salary ~ Experience + Score + Degree

(show the beta coefficients, R^2 and the first 5 values of y $(fitted.values) and$ (residuals)

```
reg <- lm(Salary ~ Experience + Score + Degree, data=programmer_salaries)
summary(reg)
```

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

```
reg$fitted.values[1:5]
```

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

```
reg$residuals[1:5]
```

```
##          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:

(i) Create an X matrix that has a first column of 1s followed by columns of the independent variables (only show the code)

```
intercept <- rep(1, 20)
x1 <- programmer_salaries$Experience
x2 <- programmer_salaries$Score
x3 <- programmer_salaries$Degree
X <- cbind(intercept, x1, x2, x3)
X
```

(ii) Create a y vector with the Salary values (only show the code)

```
y <- programmer_salaries$Salary
y
```

(iii) Compute the beta_hat vector of estimated regression coefficients (show the code and values)

```
beta_hat <- solve((t(X)%*%X))%*%t(X)%*%y
beta_hat
```

```
##                [,1]
## intercept 7.944849
## x1        1.147582
## x2        0.196937
## x3        2.280424
```

(iv) Compute a y_hat vector of estimated y values, and a res vector of residuals (show the code and the first 5 values of y_hat and res)

```
y_hat <- 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
```

(v) Using only the results from (i) – (iv), compute SSR, SSE and SST (show the code and values)

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

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

```
SSE <- sum((y-y_hat)^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 (show code and values):

(i) Use any combination of SSR, SSE, and SST

```
R <- SSR/SST
R
```

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

```
R <- 1 - SSE/SST
R
```

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

(ii) Use the squared correlation of vectors y and y

```
cor(y, y_hat)^2
```

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

Question 3) We're going to take a look back at the early heady days of global car manufacturing, when American, Japanese, and European cars competed to rule the world. Take a look at the data set in file auto-data.txt. We are interested in explaining what kind of cars have higher fuel efficiency (mpg).

- mpg: miles-per-gallon (dependent variable)
- cylinders: cylinders in engine
- displacement: size of engine
- horsepower: power of engine
- weight: weight of car
- acceleration: acceleration ability of car
- model_year: year model was released
- origin: place car was designed (1: USA, 2: Europe, 3: Japan)
- car_name: make and model names

Note that the data has missing values ('?' in data set), and lacks a header row with variable names:

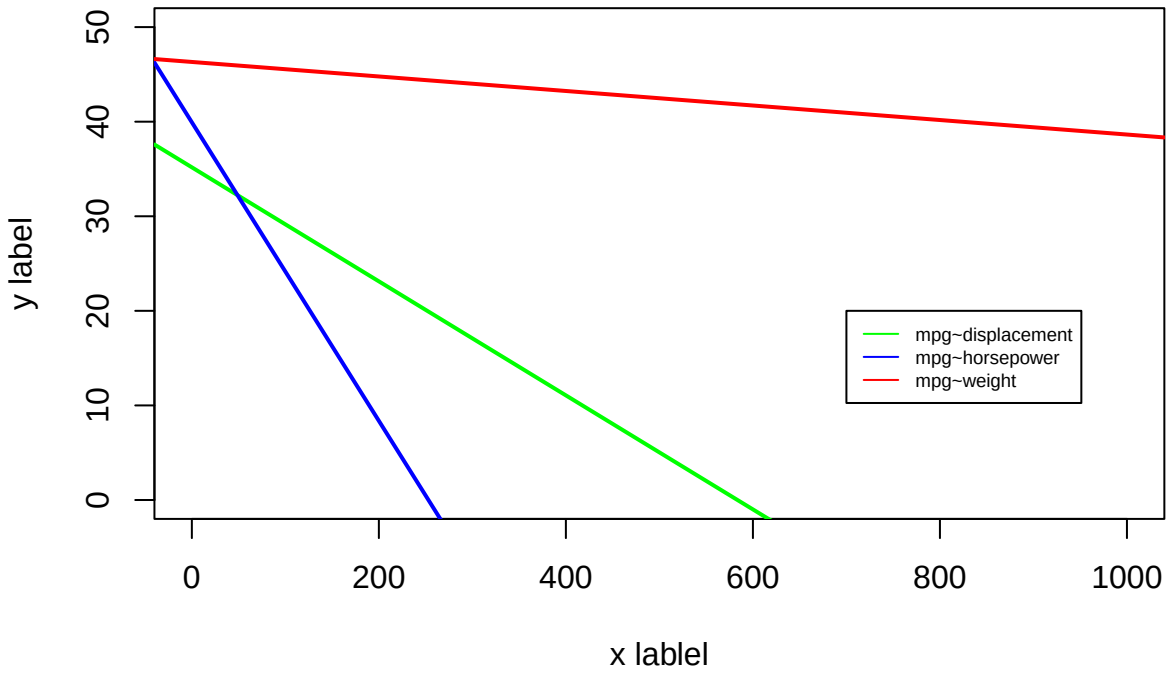
```
auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
names(auto) <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
                 "acceleration", "model_year", "origin", "car_name")
head(auto, 5)
```

```
##   mpg cylinders displacement horsepower weight acceleration model_year origin
## 1  18         8          307        130   3504         12.0         70      1
## 2  15         8          350        165   3693         11.5         70      1
## 3  18         8          318        150   3436         11.0         70      1
## 4  16         8          304        150   3433         12.0         70      1
## 5  17         8          302        140   3449         10.5         70      1
##                     car_name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
```

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

(i) Visualize the data in any way you feel relevant (report only relevant/interesting ones)

```
plot(NULL, xlim=c(0,1000), ylim=c(0,50), ylab="y label", xlab="x lablel")
lm.model1 <- lm(mpg~displacement, data=auto)
lm.model2 <- lm(mpg~horsepower, data=auto)
lm.model3 <- lm(mpg~weight, data=auto)
abline(lm.model1, lwd=2, col="green")
abline(lm.model2, lwd=2, col="blue")
abline(lm.model3, lwd=2, col="red")
legend(700, 20, legend = c("mpg~displacement", "mpg~horsepower", "mpg~weight"),
       col = c("green", "blue", "red"), lty = 1, cex = 0.6)
```



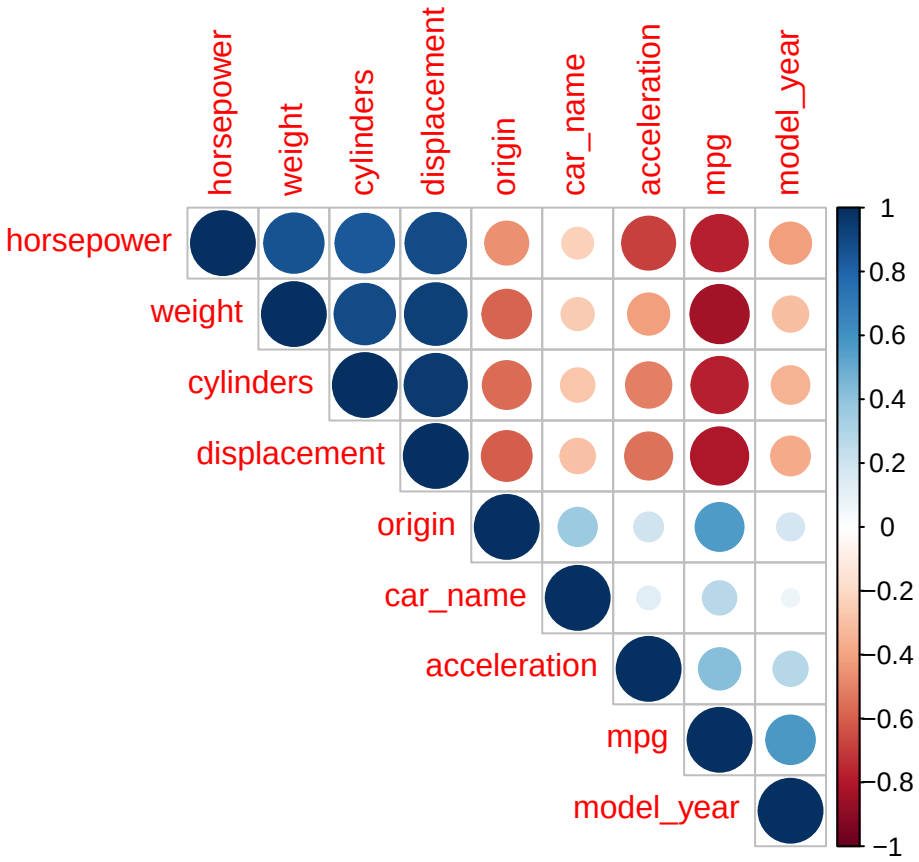
(ii) Report a correlation table of all variables, rounding to two decimal places (in the cor() function, set use="pairwise.complete.obs" to handle missing values)

```
auto_m <- data.matrix(auto)
res <- cor(auto_m, use="pairwise.complete.obs")
round(res, 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
## car_name      0.27     -0.28        -0.29      -0.23  -0.26         0.13
##              model_year origin car_name
## mpg                0.58   0.56     0.27
## cylinders         -0.35  -0.56    -0.28
## displacement      -0.37  -0.61    -0.29
## horsepower        -0.42  -0.46    -0.23
## weight            -0.31  -0.58    -0.26
## acceleration       0.29   0.21     0.13
## model_year         1.00   0.18     0.07
## origin             0.18   1.00     0.36
## car_name           0.07   0.36     1.00
```

```
corrplot(res, type="upper", order="hclust")
```



Positive correlations are displayed in blue and negative correlations in red color. Color intensity and the size of the circle are proportional to the correlation coefficients. In the right side of the correlogram, the legend color shows the correlation coefficients and the corresponding colors.

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

ans: horsepower, weight, cylinders, displacement

(iv) Which relationships might not be linear? (don't worry about linearity for rest of this HW)

ans: car_name to all the other relationship.

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

ans: Yes. Such as mpg~housepower, mpg~weight, mpg~cylinders, mpg~displacement, displacement~housepower, displacement~weight, displacement~cylinders cylinders~housepower, cylinders~weight, weight~housepower

(b) Let's create a linear regression model where mpg is dependent upon all other suitable variables (Note: origin is categorical with three levels, so use factor(origin) in lm(...) to split it into two dummy variables)

```
auto_m2 <- as.data.frame(auto_m)
lm2 <- lm(mpg ~ cylinders + displacement + horsepower+ weight + acceleration
        + model_year + factor(origin), data=auto_m2)
summary(lm2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year + factor(origin), data = auto_m2)
##
## 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 ***
## cylinders       -4.897e-01  3.212e-01  -1.524 0.128215
## displacement     2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower      -1.818e-02  1.371e-02  -1.326 0.185488
## weight          -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration     7.910e-02  9.822e-02   0.805 0.421101
## 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 ***
## ---
## 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 independent variables have a 'significant' relationship with mpg at 1% significance?

	Pr	significance
cylinders	0.128215	
displacement	0.001863	**
horsepower	0.185488	
weight	< 2e-16	***
acceleration	0.421101	
model_year	< 2e-16	***
factor(origin)2	4.72e-06	***
factor(origin)3	3.93e-07	***

ans: displacement have a 'significant' relationship with mpg at 1% significance.

(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? (hint: units!)

```
lm2$coefficients
```

```
##     (Intercept)       cylinders    displacement      horsepower          weight
##   -17.954602067    -0.489709424     0.023978644    -0.018183464    -0.006710384
##    acceleration      model_year factor(origin)2 factor(origin)3
##     0.079103036     0.777026939     2.630002360     2.853228228
```

ans: Estimate of cylinders is the largest and is significant, therefore it's the most effective one.

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

(i) Create fully standardized regression results: are these slopes easier to compare? (note: consider if you should standardize origin)

```
auto_std <- auto
auto_std[,1:7] <- lapply(auto_std[,1:7], function(x) c(scale(x)))
auto_std <- data.frame(auto_std, auto$car_name)
summary(lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration
        + model_year + factor(origin), data=auto_std))
```

```
##
## 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
```

ans: It is easier to compare since the coefficients are more or less around (1,-1).

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

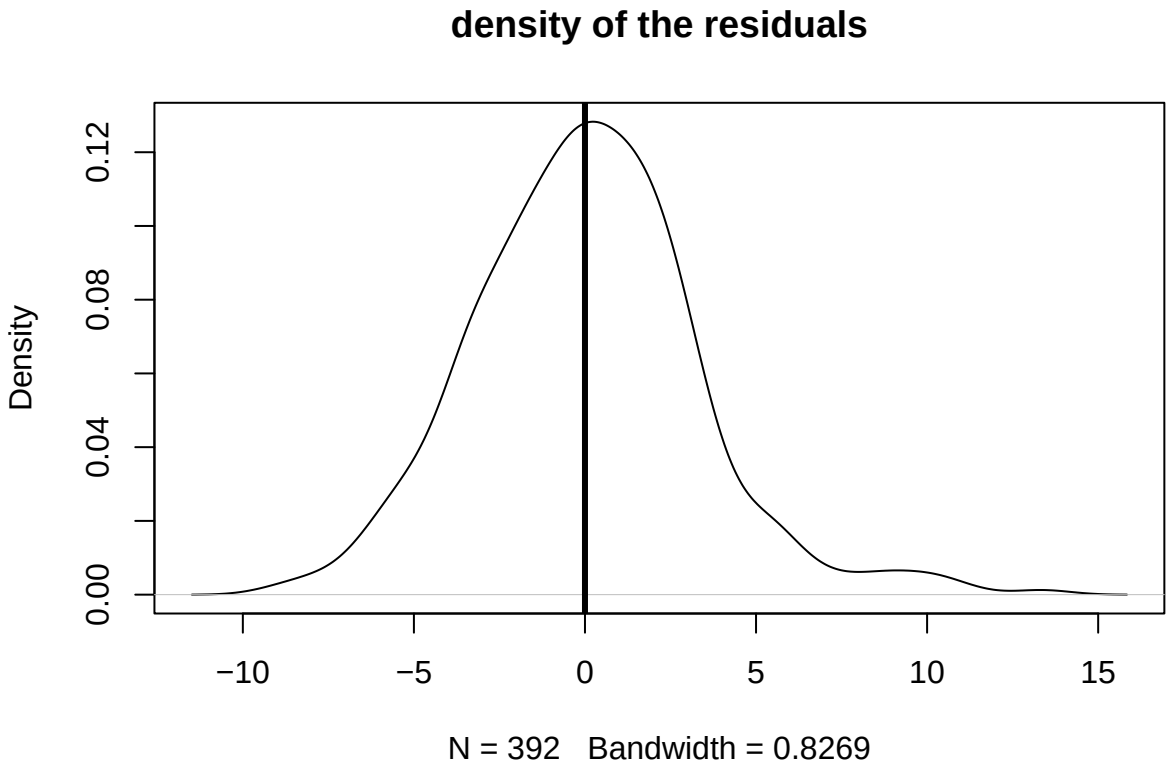
```
summary(lm(mpg ~ weight + acceleration + model_year, data=auto_m2))
```

```
##
## Call:
## lm(formula = mpg ~ weight + acceleration + model_year, data = auto_m2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7281 -2.3348 -0.1332  2.0553 14.2889
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.471e+01  4.020e+00  -3.658 0.000289 ***
## weight       -6.598e-03  2.294e-04 -28.756  < 2e-16 ***
## acceleration  5.597e-02  7.003e-02   0.799 0.424640
## model_year    7.492e-01  4.986e-02  15.025  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.436 on 394 degrees of freedom
## Multiple R-squared:  0.8082, Adjusted R-squared:  0.8067
## F-statistic: 553.3 on 3 and 394 DF,  p-value: < 2.2e-16
```

ans: The acceleration becomes significant.

(iii) Plot the density of the residuals: are they normally distributed and centered around zero? (get the residuals of a fitted linear model, e.g. regr <- lm(...), using regr$residuals

```
regr <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration
          + model_year + factor(origin), data=auto_m2)
plot(density(regr$residuals), main="density of the residuals")
abline(v=mean(regr$residuals), lwd = 3)
```



ans: They are normally distributed and centered around zero.