BACS - HW10

109006241

Question 1) We will use the interactive_regression() function from CompStatsLib again – Windows users please make sure your desktop scaling is set to 100% and RStudio zoom is 100%; alternatively, run R from the Windows Command Prompt.

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

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

   Scenario 1

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

   Scenario 3

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

   Scenario 2 has bigger SSE, SSR, and SST.

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

   Scenario 4 has bigger SSE, SSR, and SST.

Question 2) Let's analzye the programmer_salaries.txt dataset we saw in class. Read the file using read.csv("programmer_salaries.txt", sep="") because the columns are separated by tabs ().

```r
data1 = read.csv("programmer_salaries.txt", sep="\t")
```

a. Use the lm() function to estimate the regression model Salary ~ Experience + Score + Degree. Show the beta coefficients, R2, and the first 5 values of y (fitted.values) and (residuals)

```r
model = lm(Salary ~ Experience + Score + Degree, data1)
```

```r
# Beta Coefficients
model$coefficients
```

```
## (Intercept)  Experience       Score      Degree 
##    7.944849    1.147582    0.196937    2.280424
```

```r
# R-squared
summary(model)$r.squared
```

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

```r
# First 5 values of y ($fitted.values)
head(model$fitted.values, 5)
```

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

```r
# First 5 values of ($residuals)
head(summary(model)$residuals, 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)

```r
X = as.matrix(cbind(rep(1, nrow(data1)), data1[1:ncol(data1)-1]))
colnames(X)[1] = "Intercept"
```

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

```r
y = data1$Salary
```

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

```r
beta_hat = solve(t(X) %*% X) %*% t(X) %*% y
beta_hat
```

```
##                  [,1]
## Intercept  7.944849
## Experience 1.147582
## Score      0.196937
## Degree     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
res = y - y_hat
```

```
head(y_hat, 5)
```

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

```
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)
SSE = sum((y - y_hat)^2)
SST = SSR + SSE
cat(" SSR =", SSR, "\n",
     "SSE =", SSE, "\n",
     "SST =", SST)
```

```
##  SSR = 507.896
##  SSE = 91.88949
##  SST = 599.7855
```

c. Compute R2 for in two ways, and confirm you get the same results (show code and values):

   i. Use any combination of SSR, SSE, and SST

```
SSR / SST
```

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

ii. Use the squared correlation of vectors y and y_head

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

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

> Yes, we got the same values.

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")

```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")
```

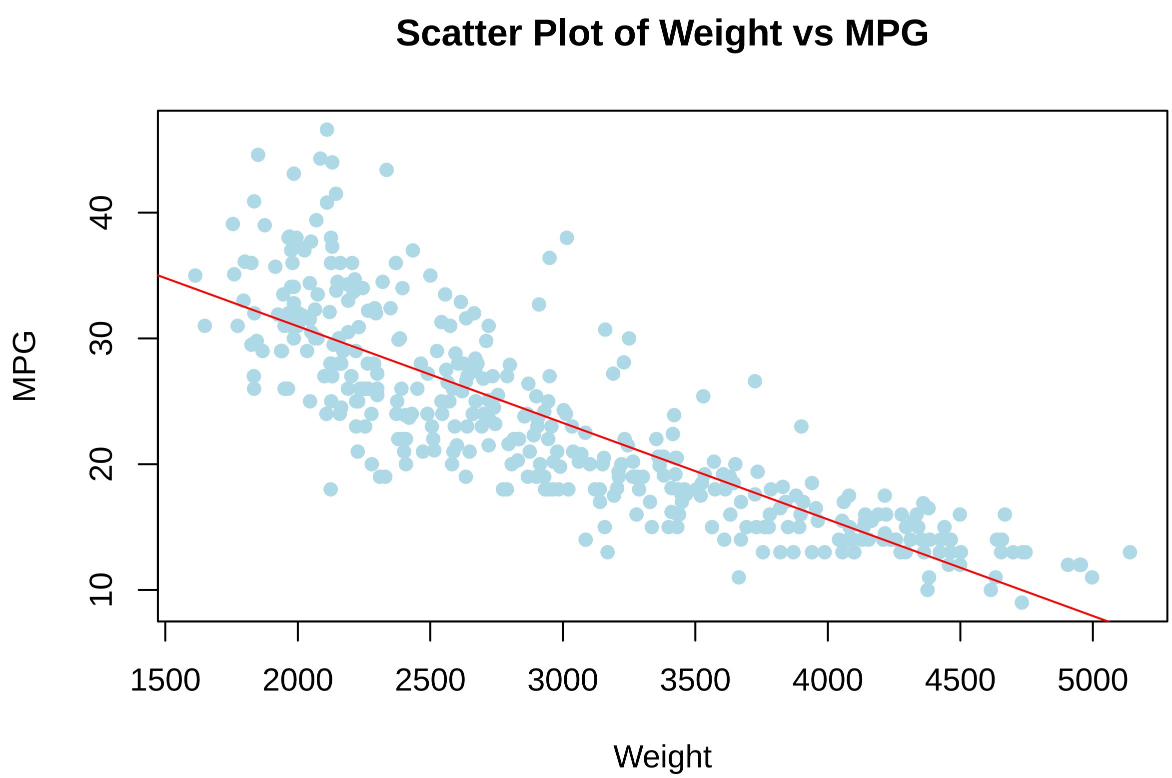
```r
head(auto, 5)
```

```
##   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 as you wish (report only relevant/interesting plots)

```
plot(auto$weight, auto$mpg,
     main="Scatter Plot of Weight vs MPG", xlab="Weight", ylab="MPG",
     pch=16, col="lightblue")
abline(lm(auto$mpg ~ auto$weight), col="red")
```



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)

```
cor_table = round(cor(subset(auto, select=-c(car_name)), use="pairwise.complete.obs"), 2)
cor_table
```

```
##                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 the visualizations and correlations, which variables appear to relate to mpg?

The "cylinders", "displacement", "horsepower", "weight" variables appear to have a significant relation to mpg.

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

The relationship between car_name and all other variables is not linear. Also, the relationship between mpg and displacement, horsepower, weight, looks more like a cone-shaped relationship than a linear relationship.

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

The "cylinders", "displacement", "horsepower", "weight" variables.

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)

i. Which independent variables have a 'significant' relationship with mpg at 1% significance?

```r
model = lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + factor(ori
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year + factor(origin), 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 ***
## 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
```

The "displacement", "weight", "model_year", "factor(origin)2" (origin = 2), and "factor(origin)3" (origin = 3) variables.

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!)

No, we are still not 100% sure about that yet. This is because the variables are not standardized yet, meaning that they are all in different unit scales.

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)

The "origin" variable doesn't need to be standardized, because it is a categorical variable.

```
variables = c("mpg", "cylinders", "displacement", "horsepower",
              "weight", "acceleration", "model_year")
auto_std = auto
auto_std[variables] = scale(auto[variables])

model = lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + factor(ori
summary(model)
```

```
##
## 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
```

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

```
summary(lm(mpg ~ cylinders, auto_std))
```

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

```
summary(lm(mpg ~ horsepower, auto_std))
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = auto_std)
##
## 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
## 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 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
```

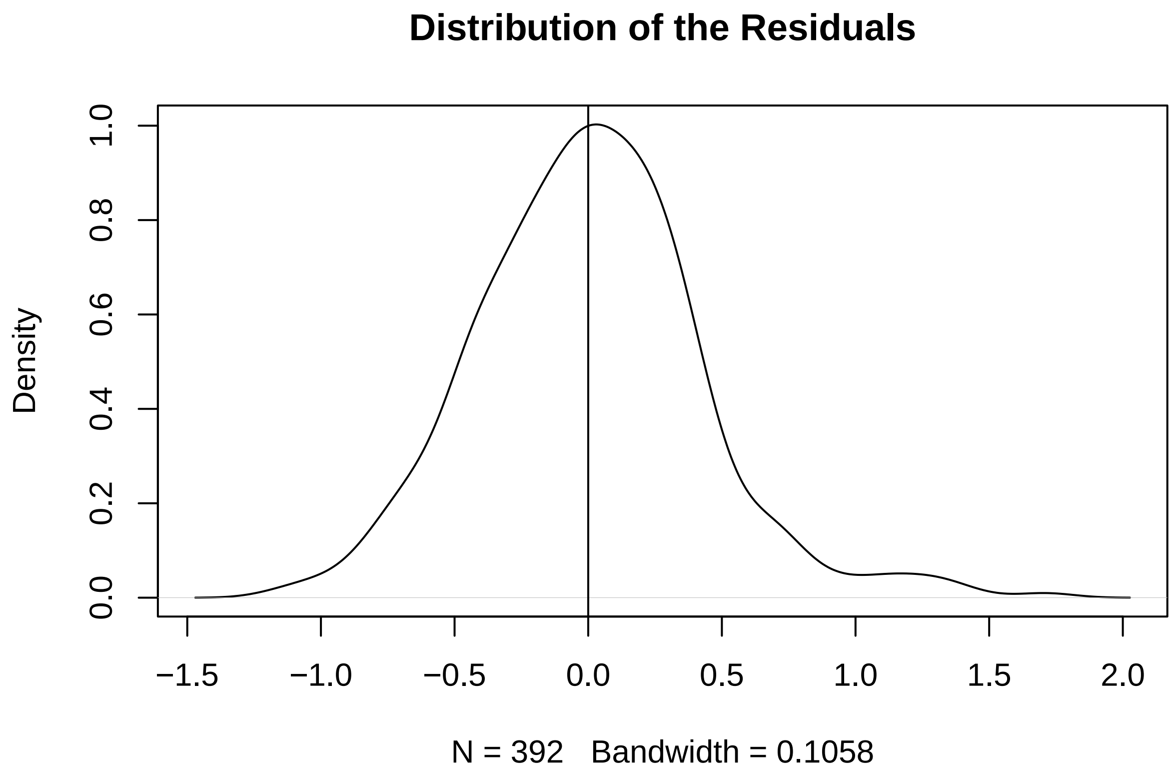
```
summary(lm(mpg ~ acceleration, auto_std))
```

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

They all become significant, because all their p-values are below 0.01.

iii. Plot the distribution 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

```
res = model$residuals
plot(density(res), main="Distribution of the Residuals")
abline(v=mean(res))
```



Yes, they seem to be normally distributed and centered around zero.