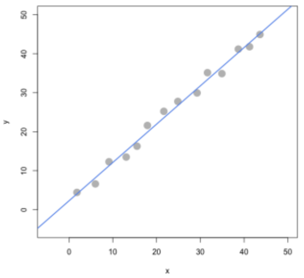
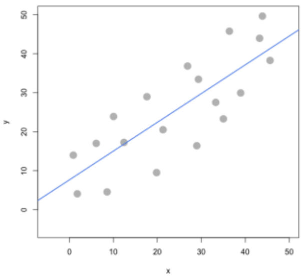
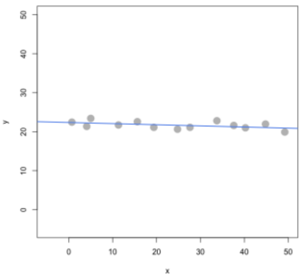
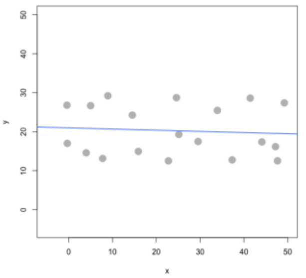
BACS - HW 10 106073401

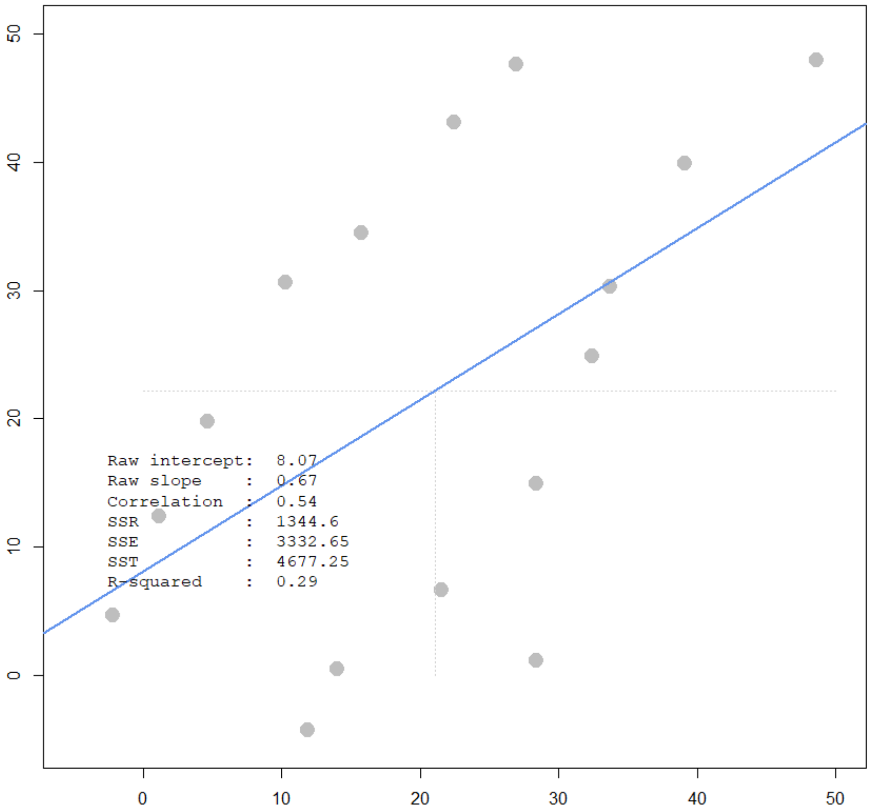
Question 1) Model fit is often determined by R^2 so let's dig into what this perspective of model fit is all about. Download `demo_simple_regression_rsq.R` from Canvas – it has a function that runs a regression simulation. This week, the simulation also reports R^2 along with the other metrics from last 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

Scenario 1	*Scenario 2*	*Scenario 3*	*Scenario 4*
			

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()`

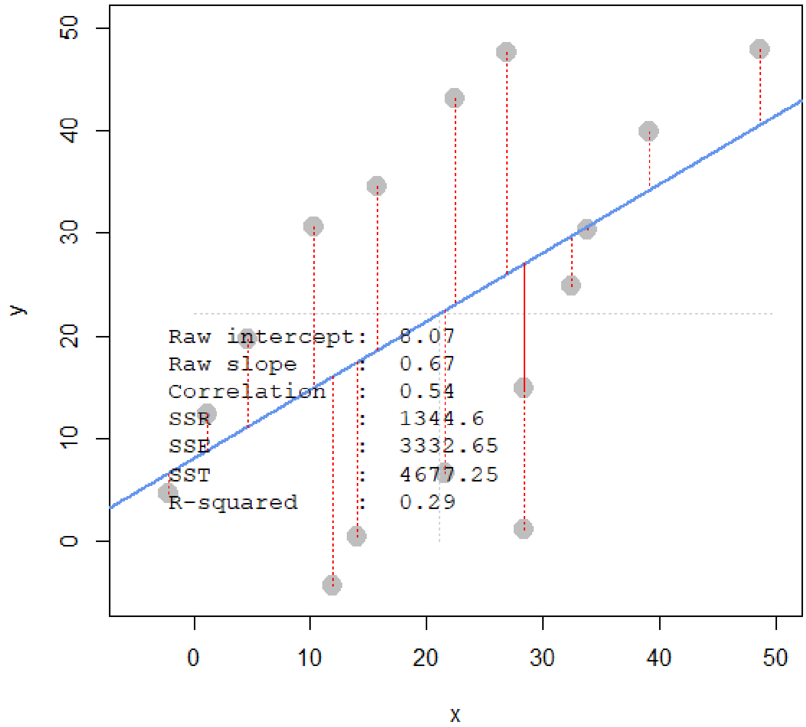
```
pts<-interactive_regression_rsq()
```



ii. Run a linear model of x and y points to confirm the R^2 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
-25.900 -13.060   1.602  10.350  21.553

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.0718     7.0748   1.141   0.2730
x             0.6691     0.2816   2.377   0.0323 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.43 on 14 degrees of freedom
Multiple R-squared:  0.2875,	Adjusted R-squared:  0.2366
F-statistic: 5.648 on 1 and 14 DF,  p-value: 0.03228
```

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



iv. Use only `pts$x`, `pts$y`, `y_hat` and `mean(pts$y)` to compute SSE, SSR and SST, and verify R^2

```
> sse <- sum((pts$y - y_hat)^2)
> sse
[1] 3332.649
> ssr <- sum((y_hat - mean(pts$y))^2)
> ssr
[1] 1344.6
> sst <- sse + ssr
> sst
[1] 4677.249
> r_square <- ssr/sst
> r_square
[1] 0.2874767
```

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

Scenario 1

c. Comparing scenarios 3 and 4, which do we expect to have a stronger R^2 ?

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)

<u>Scenario 1</u> might has <u>bigger</u> SSR,SST and smaller SSE.

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)

<u>Scenario 3</u> might has <u>smaller</u> SSR,SST and smaller SSE.

Question 2) 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 a data set (`auto-data.txt`). We are interested in explaining what kind of cars have higher fuel efficiency (measured by `mpg`).

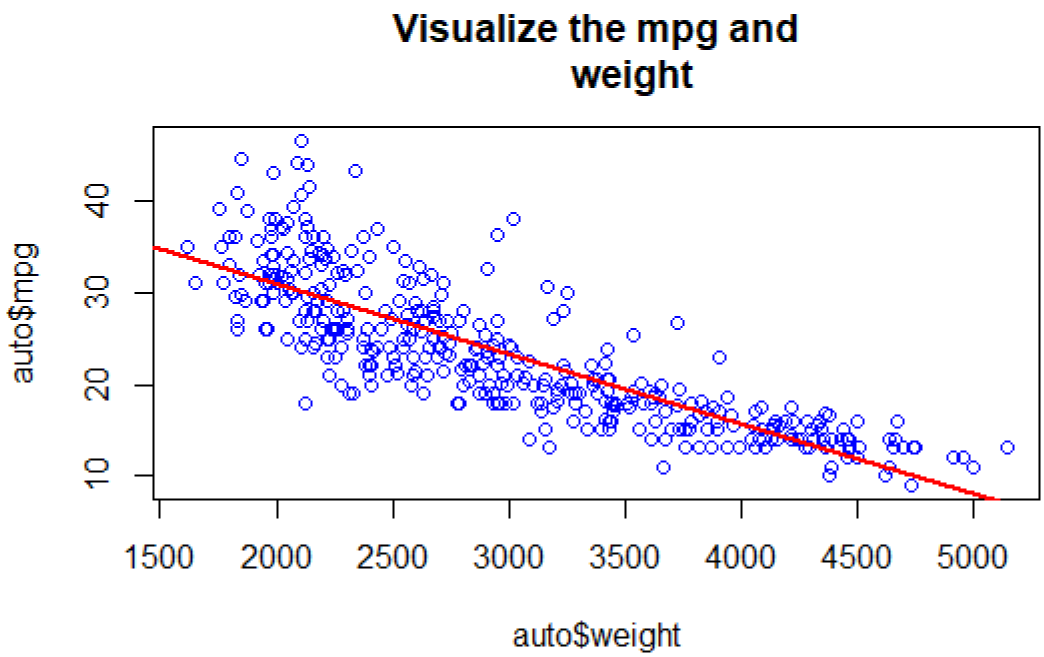
```
1. mpg:           miles-per-gallon (dependent variable)
2. cylinders:     cylinders in engine
3. displacement:  size of engine
4. horsepower:    power of engine
5. weight:        weight of car
6. acceleration:  acceleration ability of car
7. model_year:    year model was released
8. origin:     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

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

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)

```
> auto <- read.table("auto-data.txt", header=FALSE, na.strings = "?")
> names(auto) <- c("mpg", "cylinders", "displacement", "horsepower",
+                  "weight", "acceleration", "model_year", "origin", "car_name")
>
> plot(auto$mpg~auto$weight,col="blue",main = "Visualize the mpg and
+        weight")
> regr<-lm(mpg~weight,data=auto)
> abline(regr,col='red',lwd='2')
```



ii. Report a correlation table of all variables, rounding to two decimal places

```
> round(cor(auto[,-9],use="pairwise.complete.obs"),2)
               mpg cylinders displacement horsepower weight acceleration model_year origin
mpg           1.00     -0.78        -0.80      -0.78  -0.83         0.42       0.58   0.56
cylinders    -0.78      1.00         0.95       0.84   0.90        -0.51      -0.35  -0.56
displacement -0.80      0.95         1.00       0.90   0.93        -0.54      -0.37  -0.61
horsepower   -0.78      0.84         0.90       1.00   0.86        -0.69      -0.42  -0.46
weight       -0.83      0.90         0.93       0.86   1.00        -0.42      -0.31  -0.58
acceleration  0.42     -0.51        -0.54      -0.69  -0.42         1.00       0.29   0.21
model_year    0.58     -0.35        -0.37      -0.42  -0.31         0.29       1.00   0.18
origin        0.56     -0.56        -0.61      -0.46  -0.58         0.21       0.18   1.00
```

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

Weight seems to relate to mpg.

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

cylinders & model_year, weight & model_year, accreleration & model_year, accreleration & origin.

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

mpg & displacement, cylinders& displacement, cylinders & horsepower, cylinders & weight, displacement & weight.

b. Let's try an ordinary linear regression, 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 factors have a 'significant' effect on mpg at 1% significance?

```
> summary(with(auto,lm(mpg~cylinders+displacement+horsepower+weight+accelerati
on+model_year+factor(origin))))

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

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
```

Displacement, weight, model_year & origin have significant effect on mpg.

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, since all variables are measured in 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? (note: consider if you should standardize `origin`)

```
> auto_std <- data.frame(scale(auto[,c(-8,-9)]),origin=auto[,8])
> summary(with(auto_std,lm(mpg~cylinders+displacement+horsepower+weight+accelera
tion+model_year+factor(origin))))

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

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 are significant if we regress `mpg` over them individually?

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

Residuals:
     Min       1Q   Median       3Q      Max
-1.99021 -0.42496 -0.01343  0.46422  1.80240

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.256e-15  3.169e-02    0.00        1
mpg         -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(horsepower~mpg,data = auto_std))

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

Residuals:
     Min       1Q   Median       3Q      Max
-1.68590 -0.40829 -0.05439  0.34054  2.51867

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.006847   0.031747  -0.216    0.829
mpg         -0.779522   0.031831 -24.489   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6285 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(acceleration~mpg,data = auto_std))

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

Residuals:
     Min       1Q   Median       3Q      Max
-2.23273 -0.62713 -0.08554  0.52992  3.14952

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.004e-16  4.554e-02   0.000        1
mpg          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
```

Both of them are significant if we regress them individually.

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

```
> regr <-with(auto_std,lm(mpg~cylinders+displacement+horsepower+weight+accelerat
ion+model_year+factor(origin)))
> plot(density(regr$residuals))
```