HW11

108078506

5/8/2021

Question 1)

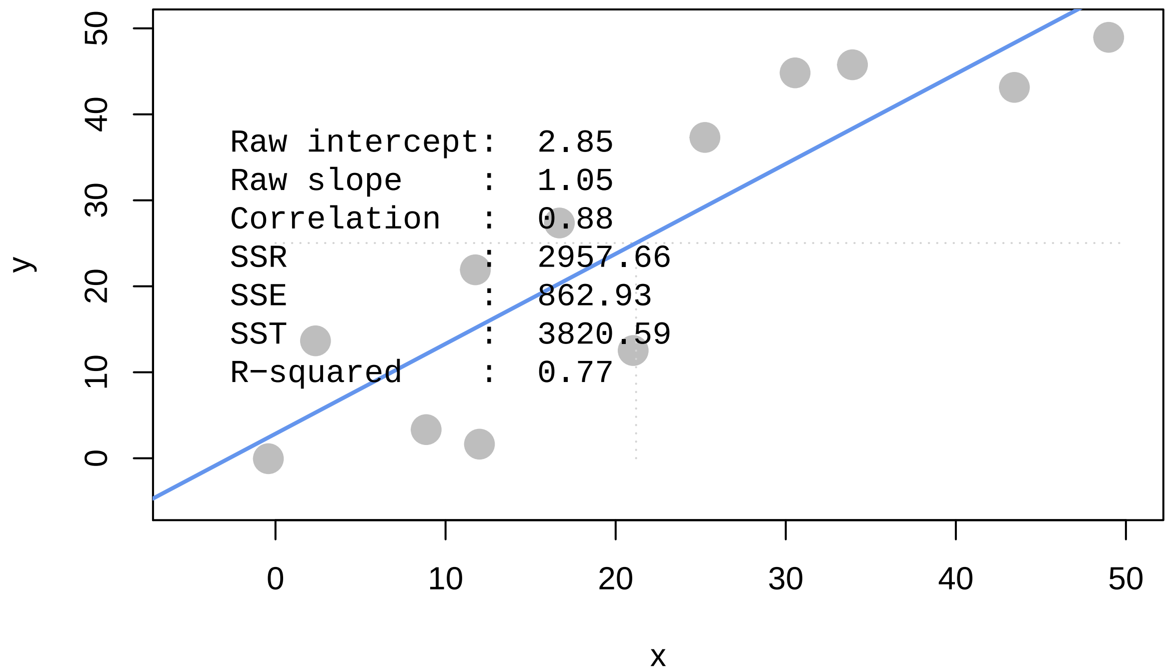
a. Let's dig into what regression is doing to compute model fit:

i. Plot Scenario 2, storing the returned points: pts <- interactive_regression_rsq()

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

```
# pts <- interactive_regression_rsq()
# Turn pts which generated from the demo_simple_regression_rsq.R into data frame
pts <- data.frame(x=c(-0.4184518, 2.3529698, 21.0299416, 16.6920643, 25.2473223,
                      48.9851509, 33.9230769, 11.9926972, 8.8597858, 30.5491723,
                      43.4423077,11.7517040)
                  , y= c(-0.05006954, 13.65646732, 12.52990264, 27.36300417, 37.31432545,
                         48.95549374, 45.76356050, 1.63977747, 3.32962448, 44.82475661,
                         43.13490960, 21.91794159))

plot_regr_rsq(pts)
```



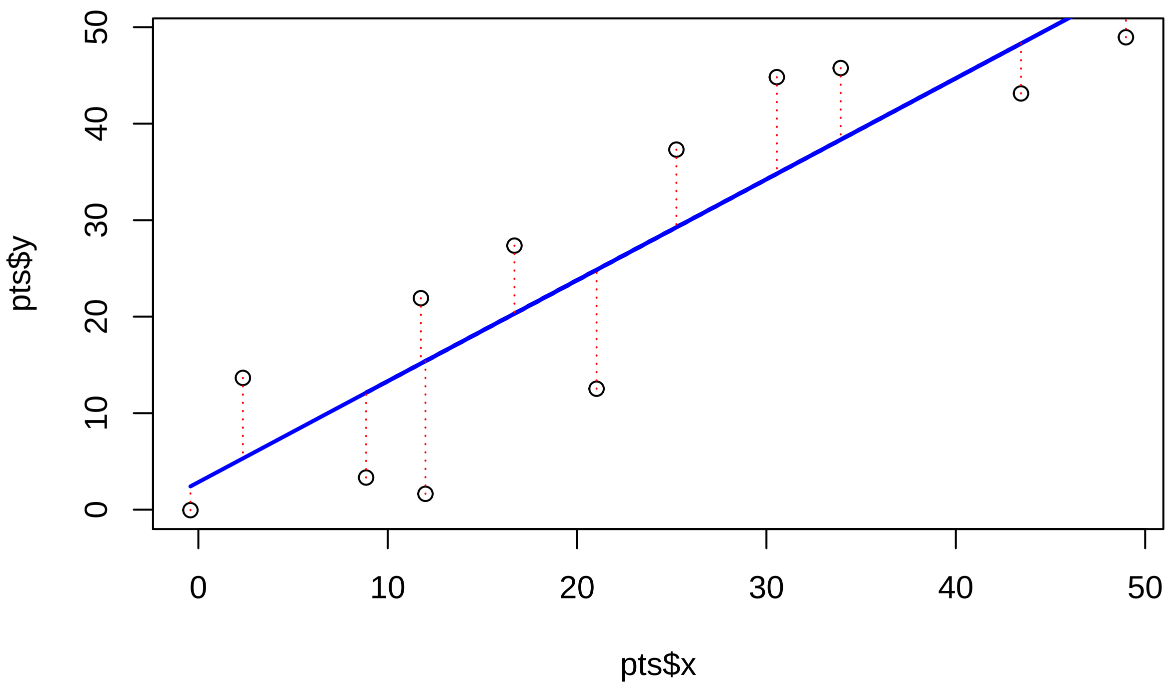
ii. Run a linear model of x and y points to confirm the R2 value reported by the simulation

```
regr <- lm(pts$y ~ pts$x)
summary(regr)
```

```
##
## Call:
## lm(formula = pts$y ~ pts$x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.758  -6.074   2.156   7.577  10.012
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.8492     4.6419   0.614 0.553055
## pts$x         1.0463     0.1787   5.854 0.000161 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.289 on 10 degrees of freedom
## Multiple R-squared:  0.7741, Adjusted R-squared:  0.7516
## F-statistic: 34.27 on 1 and 10 DF,  p-value: 0.0001606
```

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

```
y_hat <- regr$fitted.values
plot(pts$x, pts$y)
lines(x = pts$x, y = y_hat, col = "blue", lwd = 2)
segments(pts$x, pts$y, pts$x, y_hat, col="red", lty="dotted")
```



iv. Use only ptsx, ptsy, y_hat and mean(pts$y) to compute SSE, SSR and SST, and verify R2

```
pts_SSR <- sum((y_hat - mean(pts$y)) ^ 2)
pts_SST <- sum((pts$y - mean(pts$y)) ^ 2)
pts_SSE <- pts_SST - pts_SSR
pts_R_2 <- pts_SSR / pts_SST

cat("SSR = ", pts_SSR)
```

```
## SSR =  2957.663
```

```
cat("\nSSR = ", pts_SST)
```

```
##
## SSR =  3820.591
```

```
cat("\nSSR = ", pts_SSE)
```

```
##
## SSR =  862.9273
```

```
cat("\n R2 = ", pts_R_2)
```

```
##
##  R2 =  0.7741377
```

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

```
# Scenario 1 will have stronger R2 since points get much closer to the line than
# scenario 2 does.
```

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

```
# Scenario 3 will  have stronger R2 since points get much closer to the line
# than scenario 4 does.
```

d. Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST?

```
##         SSE     SSR      SST
## Sce.1   Lower   Higher   Higher
## Sce.2   Higher  Lower    Lower
```

e. Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST?

```
##         SSE     SSR      SST
## Sce.3   Lower   Higher   Higher
## Sce.4   Higher  Lower    Lower
```

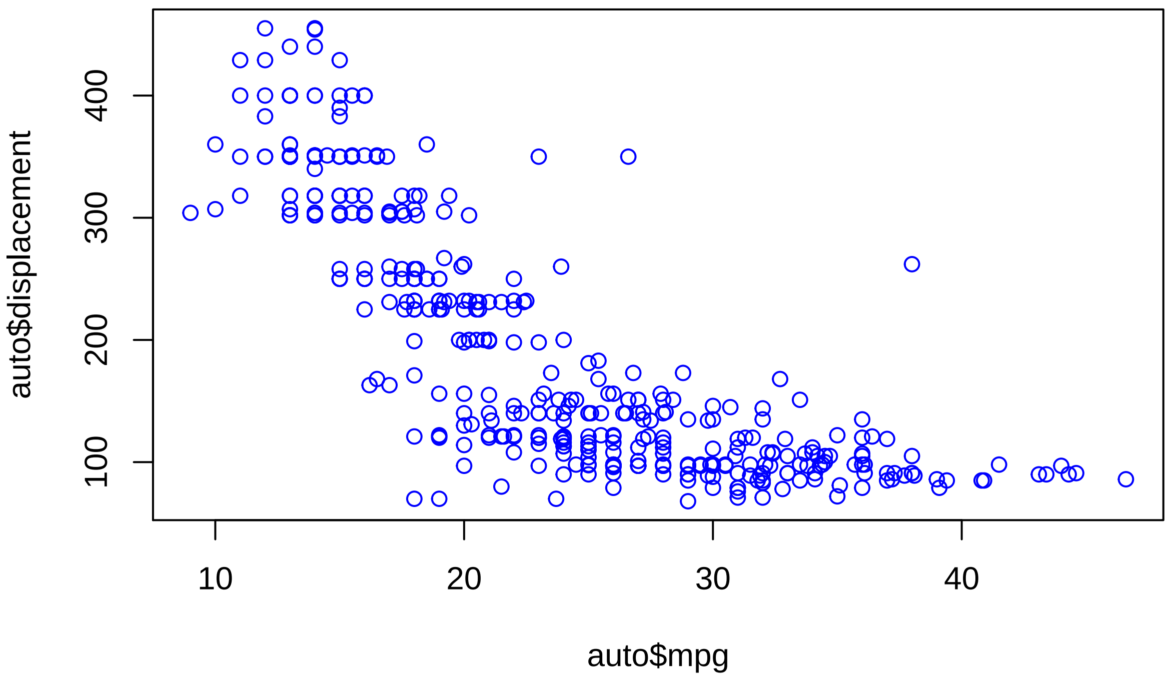
Question 2)

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(x = auto$mpg, y = auto$displacement, col = "blue")
```



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

```r
cor(auto[,1:(ncol(auto)-1)], use="pairwise.complete.obs")
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7753963   -0.8042028 -0.7784268 -0.8317409
## cylinders    -0.7753963  1.0000000    0.9507214  0.8429834  0.8960168
## displacement -0.8042028  0.9507214    1.0000000  0.8972570  0.9328241
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8317409  0.8960168    0.9328241  0.8645377  1.0000000
## acceleration  0.4202889 -0.5054195   -0.5436841 -0.6891955 -0.4174573
## model_year    0.5792671 -0.3487458   -0.3701642 -0.4163615 -0.3065643
## origin        0.5634504 -0.5625433   -0.6094094 -0.4551715 -0.5810239
##              acceleration model_year     origin
## mpg             0.4202889  0.5792671  0.5634504
## cylinders      -0.5054195 -0.3487458 -0.5625433
## displacement   -0.5436841 -0.3701642 -0.6094094
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4174573 -0.3065643 -0.5810239
## acceleration    1.0000000  0.2881370  0.2058730
## model_year      0.2881370  1.0000000  0.1806622
## origin          0.2058730  0.1806622  1.0000000
```

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

```
# According to the correlation table, displacement, horsepower and weight seem to have
# negative relation to mgp.
```

iv. Which relationships might not be linear?

```
# Origin and Mode_year have low correltion(0.1806622), so there might not be a
# linear relationship.
```

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

```
# Cylinders and displacement
# Cylinders and horsepower
# Cylinders and weight
# Displacement and horsepower
# Displacement and weight
# Horsepower and weight
```

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?

```
summary(lm(data=auto, mpg~cylinders + displacement + horsepower + weight +
             acceleration + model_year + factor(origin)))
```

```
##
## 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
```

```
# Displacement, weight, model_year and origin are lower than 1%.
```

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!)

```
# Since the units of each independent variables are different, we can't tell the most
# effective.
```

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_1 <- auto[,1:(ncol(auto)-2)]
auto_std <- data.frame(scale(auto_1))
auto_std_regr <- lm(data=auto_std, mpg~cylinders + displacement + horsepower + weight
                    + acceleration + model_year)
```

```
# After standardization, there are no difference in units of these independent variables,
# so we can compare these slopes easily.
```

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

```
summary(lm(data=auto, mpg~cylinders))
```

```
##
## Call:
## lm(formula = mpg ~ cylinders, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2607  -3.3841  -0.6478   2.5538  17.9022
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.9493     0.8330   51.56   <2e-16 ***
## cylinders    -3.5629     0.1458  -24.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.942 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
```

```r
summary(lm(data=auto, mpg~horsepower))
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 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
```

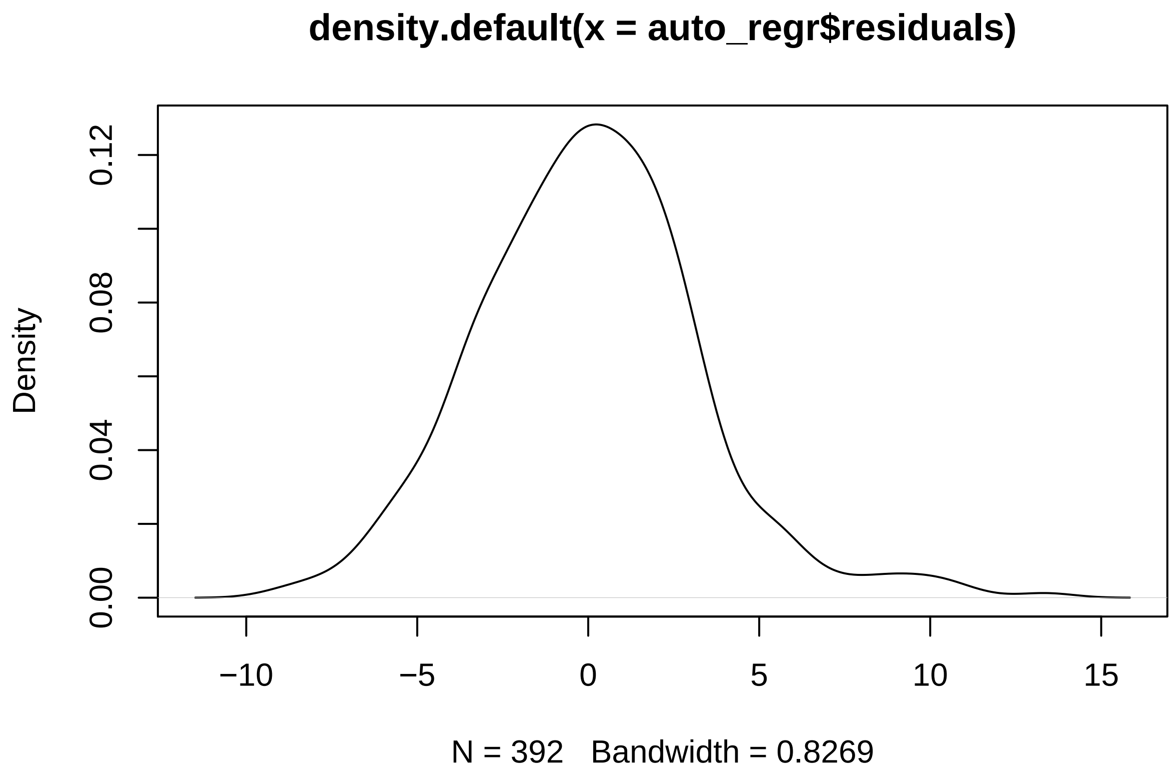
```r
summary(lm(data=auto, mpg~acceleration))
```

```
##
## Call:
## lm(formula = mpg ~ acceleration, data = auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.007  -5.636  -1.242   4.758  23.192
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.9698     2.0432   2.432   0.0154 *
## acceleration   1.1912     0.1292   9.217   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.101 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
```

```r
# All the nonsignificant independent variable become significant after we regress mpg
# over them.
```

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

```r
auto_regr <- lm(data=auto, mpg~cylinders + displacement + horsepower + weight +
             acceleration + model_year + factor(origin))
plot(density(auto_regr$residuals))
```




```
# According to the plot, the residuals are normally distributed and centered around zero.
```