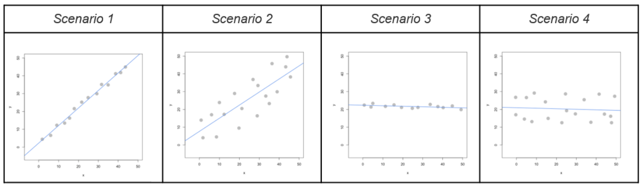
BACS_HW_Week11_106071041

106071041

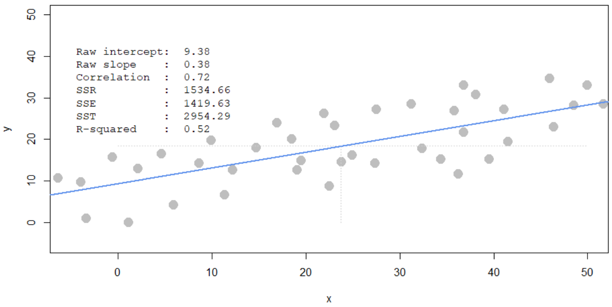
2021/5/9



Question 1

a. Scenario 2

(i) plot scenario 2 using `pts <- interactive_regression_rsq()`



(ii) Develop squared R using `regr <- lm(y ~ x, data = pts)`

```
regr <- lm(y ~ x, data = pts)
```

```
summary(regr)
```

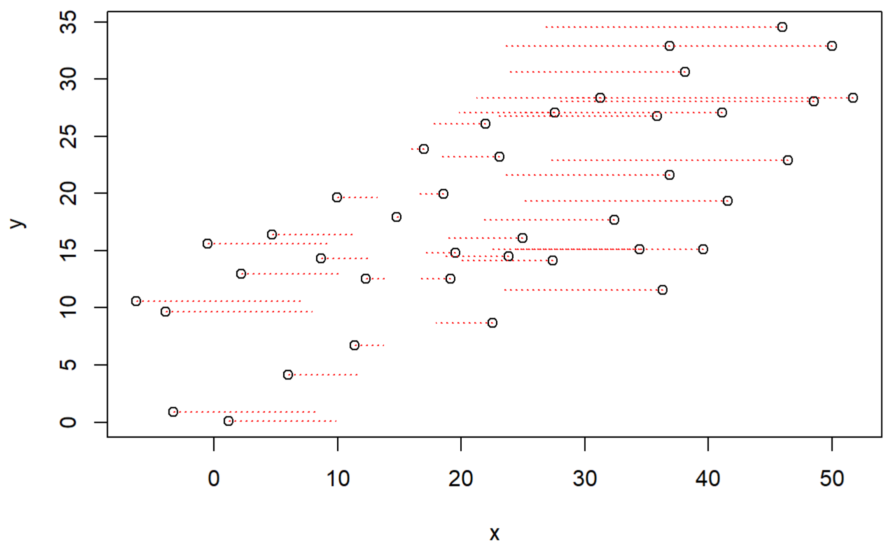
```
##
## Call:
## lm(formula = y ~ x, data = pts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5809  -4.4942   0.9461   5.1073   9.5456
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.37573    1.70643   5.494 2.81e-06 ***
## x            0.37966    0.05924   6.409 1.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.112 on 38 degrees of freedom
## Multiple R-squared:  0.5195, Adjusted R-squared:  0.5068
## F-statistic: 41.08 on 1 and 38 DF,  p-value: 1.565e-07
```

$$R^2 = 0.5195$$

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

```
y_hat <- regr$fitted.values
```

```
plot(pts)
segments(pts$x, pts$y, y_hat, col = "red", lty = "dotted")
```



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

```
actual_y <- pts$y
```

```
SSE <- sum((actual_y - y_hat)^2)
SSR <- sum((y_hat - mean(actual_y))^2)
SST <- sum((actual_y - mean(actual_y))^2)
R2 <- SSR / SST
```

$$SSE = 1419.633$$

$$SSR = 1534.659$$

$$SST = 2954.292$$

$$R^2 = 0.5195(Verified)$$

b. Scenario 1 v.s. Scenario 2: Who has stronger squared R?

Scenario 1.

c. Scenario 3 v.s. Scenario 4: Who has stronger squared R?

Scenario 3.

d. Scenario 1 v.s. Scenario 2: Compare SSE, SSR and SST

SSE: 2 > 1
SSR: Can't Sure
SST: Can't Sure

e. Scenario 3 v.s. Scenario 4: Compare SSE, SSR and SST

SSE: 4 > 3
SSR: more or less the same
SST: 4 > 3, since the difference from SSR is small

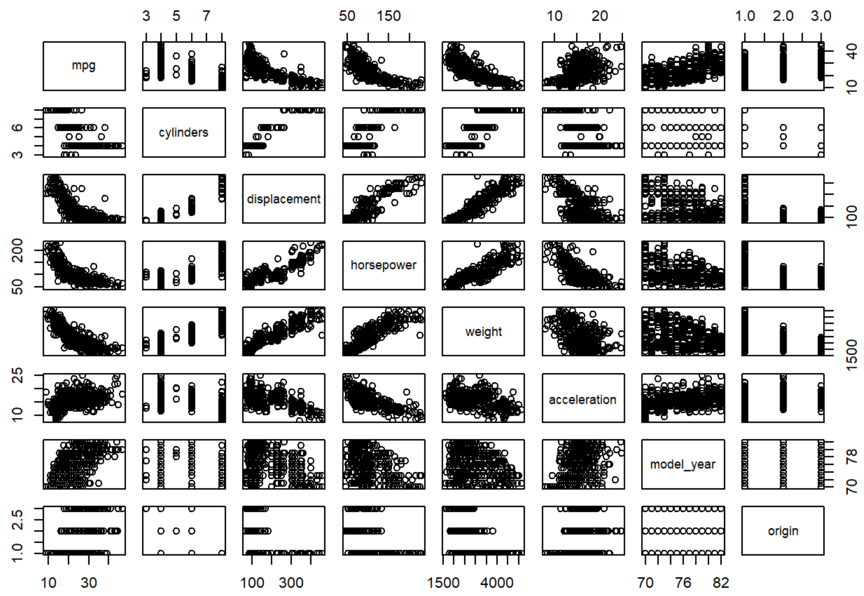
Question 2

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

a. Explore the data and problems

(i) Visualization

```
plot(auto[1:8])
```



(ii) Report a correlation table of all variables

```
cor_df <- as.data.frame(cor(auto[1:8], use = "pairwise.complete.obs"))
```

```
cor_df
```

```
##                     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) which variables seem to relate to mpg

If “related” means < -0.5 | > 0.5, then:
“cylinders”, “displacement”, “horsepower”, “weight”, “model_year”, “origin”

(iv) Which relationships might not be linear?

origin"

(v) highly correlated (r > 0.7)?

No one. (r > 0.7)
< -0.7: “cylinders”, “displacement”, “horsepower”, “weight”

```
rownames(cor_df["mpg"])[cor_df["mpg"] > 0.7]
```

```
## [1] "mpg"
```

```
rownames(cor_df["mpg"])[cor_df["mpg"] < -0.7]
```

```
## [1] "cylinders"    "displacement" "horsepower"   "weight"
```

b. linear regression model with `factor(origin)` in `lm(...)`

```
auto_regr <- lm( mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + factor(origin), data=aut
o)
summary(auto_regr)
```

```
##
## 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
```

(i) Which independent variables have a ‘significant’ relationship with mpg at 1% significance?

displacement, weight, model_year, factor(origin)2, factor(origin)3

(ii) Is it possible to determine which independent variables are the most effective at increasing mpg? If so, which ones, and if not, why not?

Yes, it’s possible. origin would be the most effective one.

c. standardization

(i) Create fully standardized regression results: are these slopes easier to compare?

I think we should not standardize origin since they are in nominal scale.
After standardization, the slopes are easier to compare.

```
standardized_auto <- as.data.frame(scale(auto[1:7]))
```

```
combine <- cbind(standardized_auto,auto[8])
```

```
head(combine)
```

```
##          mpg cylinders displacement horsepower    weight acceleration
## 1 -0.7055507  1.496308     1.089233  0.6632851 0.6300768    -1.293870
## 2 -1.0893795  1.496308     1.501624  1.5725848 0.8532590    -1.475181
## 3 -0.7055507  1.496308     1.194728  1.1828849 0.5497785    -1.656492
## 4 -0.9614365  1.496308     1.060461  1.1828849 0.5462359    -1.293870
## 5 -0.8334936  1.496308     1.041280  0.9230850 0.5651296    -1.837804
## 6 -1.0893795  1.496308     2.259274  2.4299245 1.6184551    -2.019115
##   model_year origin
## 1  -1.625381      1
## 2  -1.625381      1
## 3  -1.625381      1
## 4  -1.625381      1
## 5  -1.625381      1
## 6  -1.625381      1
```

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

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year + factor(origin), data = combine)
##
## 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) Which ones become significant when we regress mpg over them individually?

All of them become significant.

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

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

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

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

(iii) Plot the density of the residuals: are they normally distributed and centered around zero?

From the density plot, it seems that it's a normal distribution and centered around 0.

```
regr_ex <- lm( mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year + factor(origin), data = com
bine)
```

```
plot(density(regr_ex$residuals))
```