BACS HW (Week11)

106070038

2021-05-09

Question 1 Model fit is often determined by R2 so let's dig into what this perspective of model fit is all about.

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

i. Plot Scenario 2, storing the returned points

```
plot_regr_rsq <- function(points) {
  max_x <- 50
  if (nrow(points) == 0) {
    plot(NA, xlim=c(-5,max_x), ylim=c(-5,max_x), xlab="x", ylab="y")
    return()
  }
  plot(points, xlim=c(-5,max_x), ylim=c(-5,max_x), pch=19, cex=2, col="gray")
  if (nrow(points) < 2) return()

  mean_x <- mean(points$x)
  mean_y <- mean(points$y)
  segments(0, mean_y, max_x, mean_y, lwd=1, col="lightgray", lty="dotted")
  segments(mean_x, 0, mean_x, mean_y, lwd=1, col="lightgray", lty="dotted")
  regr <- lm(points$y ~ points$x)
  abline(regr, lwd=2, col="cornflowerblue")

  regr_summary <- summary(regr)
  ssr <- sum((regr$fitted.values - mean(points$y))^2)
  sse <- sum((points$y - regr$fitted.values)^2)
  sst <- sum((points$y - mean(points$y))^2)

  par(family="mono")
  legend("topleft", legend = c(
    paste(" Raw intercept: ", round(regr$coefficients[1], 2), "\n",
          "Raw slope    : ", round(regr$coefficients[2], 2), "\n",
          "Correlation  : ", round(cor(points$x, points$y), 2), "\n",
          "SSR          : ", round(ssr, 2), "\n",
          "SSE          : ", round(sse, 2), "\n",
          "SST          : ", round(sst, 2), "\n",
          "R-squared    : ", round(regr_summary$r.squared, 2))),
    bty="n")
  par(family="sans")
}
```

```
interactive_regression_rsq <- function(points=data.frame()) {
  cat("Click on the plot to create data points; hit [esc] to stop")
  repeat {
    plot_regr_rsq(points)
    click_loc <- locator(1)
    if (is.null(click_loc)) break
    if(nrow(points) == 0 ) {
      points <- data.frame(x=click_loc$x, y=click_loc$y)
    } else {
      points <- rbind(points, c(click_loc$x, click_loc$y))
    }
  }
  return(points)
}

# pts <- interactive_regression_rsq()
# save(pts, file='/Users/nkust/Desktop/2021Spring_Courses/BACS/HW9/pts.Rda')
```

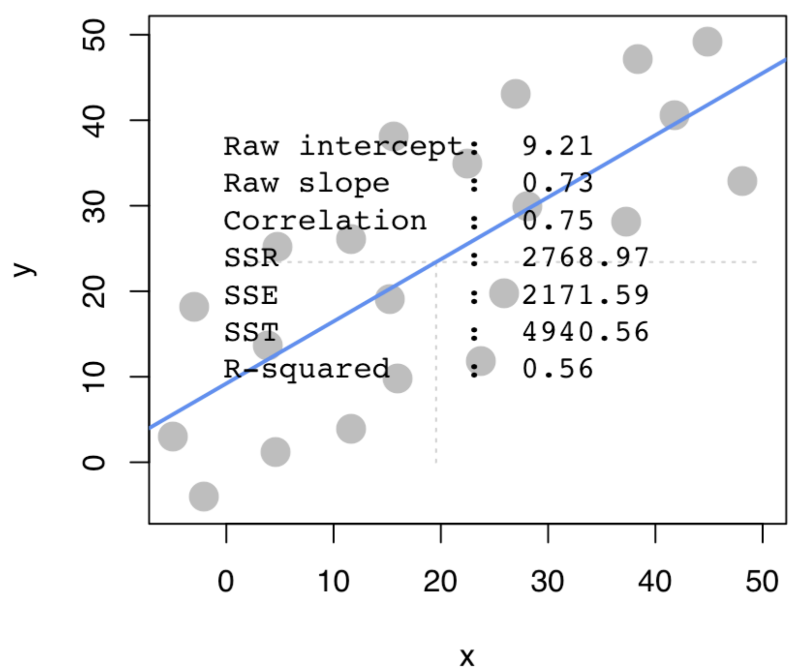


Figure 1: pts plot

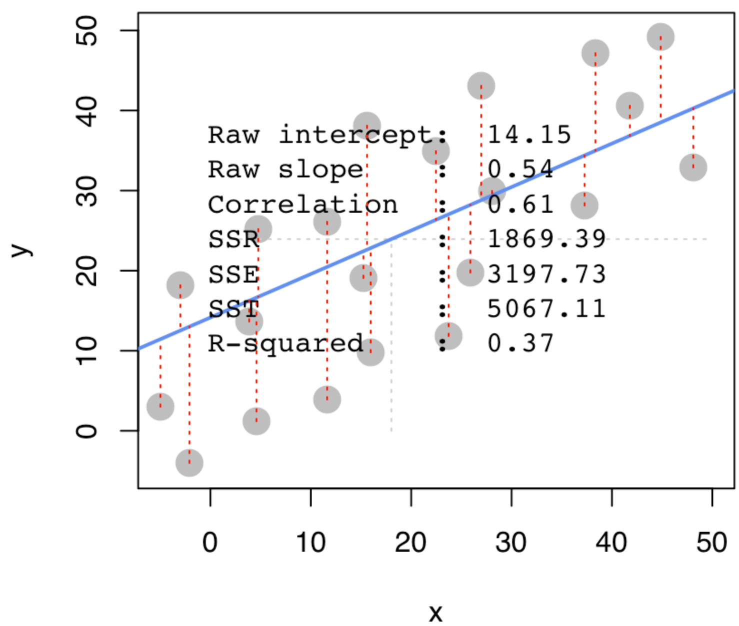
ii. Run a linear model of x and y points to confirm the R2 value reported by the simulation

```
load('/Users/nkust/Desktop/2021Spring_Courses/BACS/HW9/pts.Rda')
```

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

```
##
## Call:
## lm(formula = y ~ x, data = pts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.026  -8.449  -1.022   8.548  28.486
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.1505     3.9322   3.599  0.00179 **
## x             0.5431     0.1588   3.419  0.00272 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.64 on 20 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3374
## F-statistic: 11.69 on 1 and 20 DF,  p-value: 0.002717
```

iii. Add line segments to the plot to show the regression residuals (er-



rors)

```r
y_hat <- regr$fitted.values
# segments(pts$x, pts$y, pts$x, y_hat, col="red", lty="dotted")
```

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

```r
SSE <- sum((y_hat - mean(pts$y))^2)
SSE
```

```
## [1] 1869.386
```

```r
SSR <- sum((pts$y - y_hat)^2)
SSR
```

```
## [1] 3197.729
```

```r
SST <- SSE + SSR
SST
```

```
## [1] 5067.114
```

```r
R_square <- 1 - (SSR/(SSE + SSR))
R_square
```

```
## [1] 0.3689251
```

```r
## verify
summary(regr)$r.square
```

```
## [1] 0.3689251
```

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

- Ans: Scenario 1, because Scenarios 1 is more intensive and close to the line than Scenario 2.

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

- Ans: Scenario 3, because Scenarios 3 is more intensive and close to the line than Scenario 4.

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)

- Ans:
  - SSE: Scenario 1 $<$ Scenario 2
  - SSR: Scenario 1 $>$ Scenario 2
  - SST: Scenario 1 $\approx$ Scenario 2

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)

- Ans:
  - SSE: Scenario 3 < Scenario 4
  - SSR: Scenario 3 < Scenario 4
  - SST: Scenario 3 < Scenario 4

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 the data set in file auto-data.txt. We are interested in explaining what kind of cars have higher fuel efficiency (mpg).

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

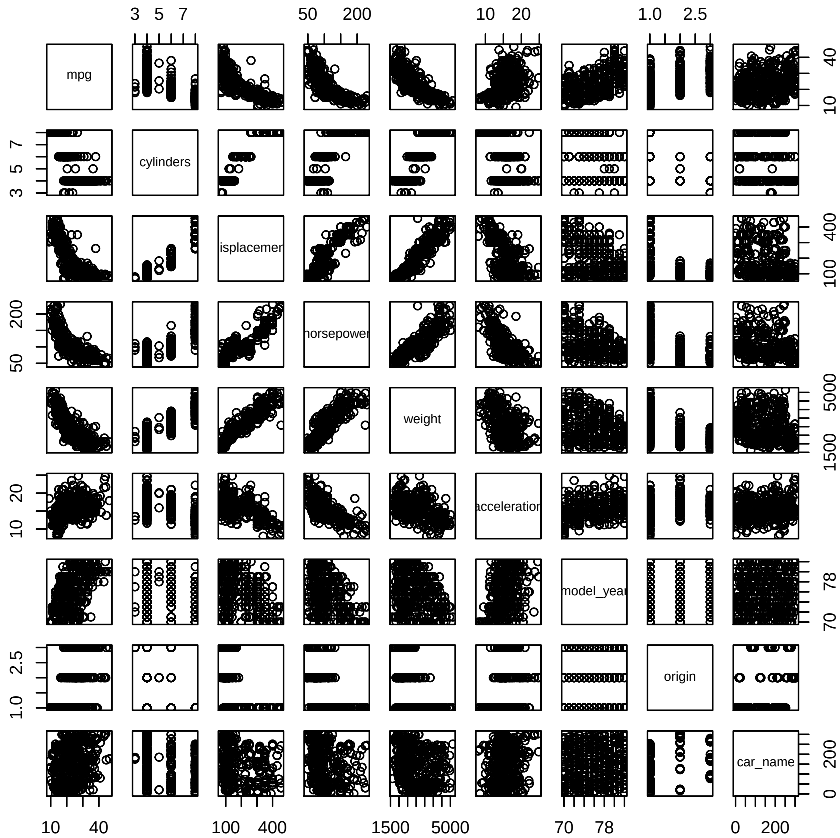
a.

i. Visualize the data in any way you feel relevant

```
summary(auto)
```

```
##       mpg          cylinders      displacement     horsepower        weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
##  1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0   1st Qu.:2224
##  Median :23.00   Median :4.000   Median :148.5   Median : 93.5   Median :2804
##  Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5   Mean   :2970
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd Qu.:3608
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
##                                                  NA's   :6
##   acceleration     model_year        origin        car_name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   Length:398
##  1st Qu.:13.82   1st Qu.:73.00   1st Qu.:1.000   Class :character
##  Median :15.50   Median :76.00   Median :1.000   Mode  :character
##  Mean   :15.57   Mean   :76.01   Mean   :1.573
##  3rd Qu.:17.18   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :24.80   Max.   :82.00   Max.   :3.000
##
```

```
plot(auto)
```



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

```
cor <- cor(auto[-9], use="pairwise.complete.obs")
round(cor,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
##              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 seem to relate to mpg?

```
cor(auto[1], auto[-9], use="pairwise.complete.obs")
```

```
##     mpg  cylinders displacement horsepower     weight acceleration model_year
## mpg   1 -0.7753963   -0.8042028 -0.7784268 -0.8317409    0.4202889  0.5792671
##        origin
## mpg 0.5634504
```

- Ans: displacement(-0.8042028) and weight(-0.8317409)

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

- Ans: Based on the correlation plot, most of the relationships with model_year are not linear.

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

- Ans: cylinders-displacement(0.95), cylinders-horsepower(0.84), cylinders-weight(0.90), displacement-horsepower(0.90), displacement-weight(0.93), horsepower-weight(0.86)

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?

```
auto$origin <- factor(auto$origin)
auto <- auto[-9]
regr <- lm(auto$mpg ~ auto$cylinders + auto$displacement + auto$horsepower + auto$weight + auto$accelera
summary(regr)
```

```
##
## Call:
## lm(formula = auto$mpg ~ auto$cylinders + auto$displacement +
##     auto$horsepower + auto$weight + auto$acceleration + auto$model_year +
##     auto$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 ***
## auto$cylinders     -4.897e-01  3.212e-01  -1.524 0.128215
## auto$displacement   2.398e-02  7.653e-03   3.133 0.001863 **
## auto$horsepower    -1.818e-02  1.371e-02  -1.326 0.185488
## auto$weight        -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## auto$acceleration   7.910e-02  9.822e-02   0.805 0.421101
## auto$model_year     7.770e-01  5.178e-02  15.005  < 2e-16 ***
## auto$origin2        2.630e+00  5.664e-01   4.643 4.72e-06 ***
## auto$origin3        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
```

- Ans: displacement

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!)

- Ans: model_year, because the Estimate of model_year is biggest(7.770e-01)

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?

```
auto_std <- data.frame(scale(auto[-8]))
auto_std_regr <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model_year , 
summary(auto_std_regr)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + model_year, data = auto_std)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.11217 -0.30532 -0.01025  0.25961  1.83734
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0004236  0.0222112   0.019    0.985
## cylinders    -0.0717877  0.0722763  -0.993    0.321
## displacement  0.1024348  0.0981565   1.044    0.297
## horsepower   -0.0019273  0.0681403  -0.028    0.977
## weight       -0.7361794  0.0725952 -10.141   <2e-16 ***
## acceleration  0.0300867  0.0360009   0.836    0.404
## model_year    0.3564069  0.0248929  14.318   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4395 on 385 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.8093, Adjusted R-squared:  0.8063
## F-statistic: 272.2 on 6 and 385 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?

- nonsignificant independent variable: cylinders, displacement, horsepower, acceleration

```
auto_std_regr_cylinders <- lm(mpg ~ cylinders , data = auto_std)
summary(auto_std_regr_cylinders)
```

```
##
## 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
```

```
auto_std_regr_displacement <- lm(mpg ~ displacement , data = auto_std)
summary(auto_std_regr_displacement)
```

```
##
## Call:
## lm(formula = mpg ~ displacement, data = auto_std)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65750 -0.39111 -0.06305  0.29782  2.38220
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.931e-17  2.983e-02    0.00        1
## displacement -8.042e-01  2.987e-02  -26.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5951 on 396 degrees of freedom
## Multiple R-squared:  0.6467, Adjusted R-squared:  0.6459
## F-statistic:   725 on 1 and 396 DF,  p-value: < 2.2e-16
```

```
auto_std_regr_horsepower <- lm(mpg ~ horsepower , data = auto_std)
summary(auto_std_regr_horsepower)
```

```
##
## 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
```

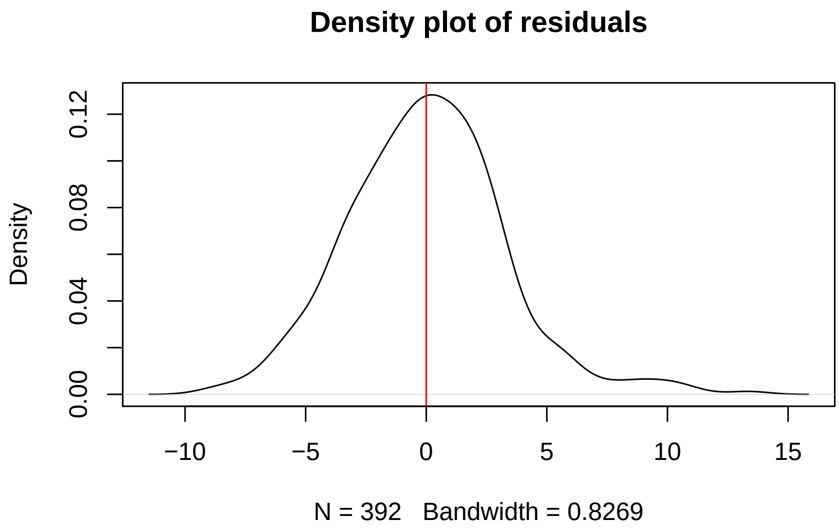
```
auto_std_regr_acceleration <- lm(mpg ~ acceleration , data = auto_std)
summary(auto_std_regr_acceleration)
```

```
##
## 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
```

- Ans: all become significant when we regress mpg over them individually (cylinders, displacement, horsepower, acceleration)

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

```
plot(density(regr$residuals), main = "Density plot of residuals")
abline(v = mean(regr$residuals), col="red")
```



- Ans: not normally distributed, but it is centered around zero