

BACS HW (Week 10)

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.

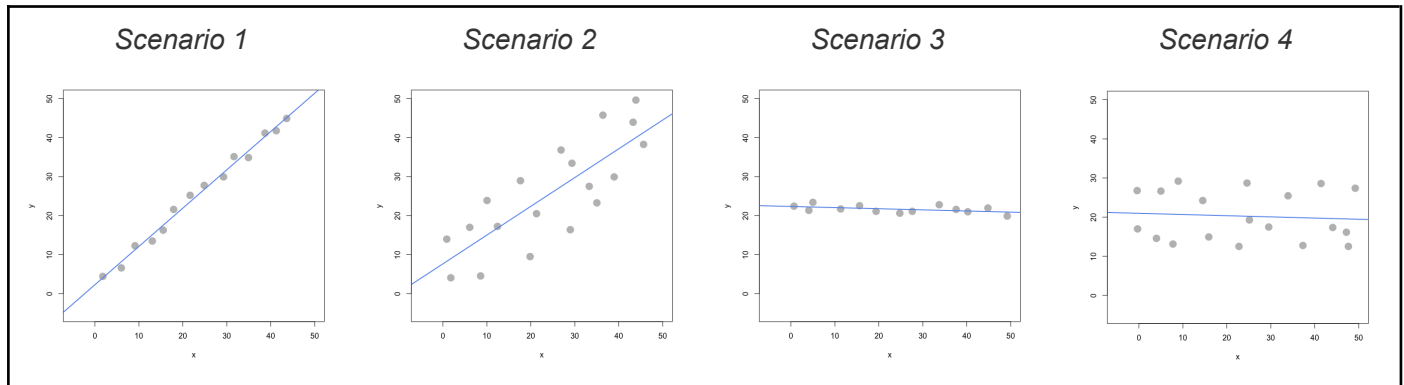
To answer the questions below, understand each of these four scenarios by simulating them:

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



- Comparing scenarios 1 and 2, which do we expect to have a stronger R^2 ?
- Comparing scenarios 3 and 4, which do we expect to have a stronger R^2 ?
- Comparing scenarios 1 and 2, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)
- Comparing scenarios 3 and 4, which do we expect has bigger/smaller SSE, SSR, and SST? (intuitively)

Question 2) Let's analyze the `programmer_salaries.txt` dataset we saw in class. Read the file using `read.csv("programmer_salaries.txt", sep="\t")` because the columns are separated by tabs (`\t`).

- Use the `lm()` function to estimate the regression model `Salary ~ Experience + Score + Degree`
Show the beta coefficients, R^2 , and the first 5 values of \hat{y} (`$fitted.values`) and ϵ (`$residuals`)
- Use only linear algebra and the geometric view of regression to estimate the regression yourself:
 - Create an X matrix that has a first column of 1s followed by columns of the independent variables (*only show the code*)
 - Create a y vector with the Salary values (*only show the code*)
 - Compute the `beta_hat` vector of estimated regression coefficients (*show the code and values*)
 - Compute a `y_hat` vector of estimated \hat{y} values, and a `res` vector of residuals (*show the code and the first 5 values of `y_hat` and `res`*)
 - Using only the results from (i) – (iv), compute SSR, SSE and SST (*show the code and values*)
- Compute R^2 for in two ways, and confirm you get the same results (*show code and values*):
 - Use any combination of SSR, SSE, and SST
 - Use the squared correlation of vectors y and \hat{y}

(see question 3 on next page)

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).

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

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")
```

- a. Let's first try exploring this data and problem:
 - i. Visualize the data as you wish (report only relevant/interesting plots)
 - 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)
 - iii. From the visualizations and correlations, which variables appear to relate to mpg?
 - iv. Which relationships might not be linear? (*don't worry about linearity for rest of this HW*)
 - v. Are there any pairs of independent variables that are highly correlated ($r > 0.7$)?
- 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?
 - 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!)
- 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)
 - ii. Regress mpg over each *nonsignificant* independent variable, individually.
Which ones become significant when we regress mpg over them individually?
 - 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`)