BIOST 546 HW 1

My-Anh Doan

2023-01-09

Q1. In this problem, we will make use of the data set `Medical_Cost_2.RData`.

  (a) Load the data set with the command `load` and check if there are missing data.
  (b) If any, remove the missing data using the command `na.omit`.

```
# load data to environment
load("./dataset/Medical_Cost_2.RData")
str(df)
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...

# check for missing data in `df`; if returned value is non-zero, there are missing data
sum(is.na(df) == TRUE)
## [1] 60

# remove the missing data from `df`
df <- na.omit(df)
str(df)
## 'data.frame':    1278 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
##  - attr(*, "na.action")= 'omit' Named int [1:60] 22 37 39 40 84 111 129 193 248 252 ...
##   ..- attr(*, "names")= chr [1:60] "22" "37" "39" "40" ...
```

  (c) We decide to focus on the outcome variable `charges` (individual medical costs billed by health insurance) and the predictors `bmi` (body mass index) and `smoker` (whether the subjects is a smoker or not). Make a scatter plot with `bmi` on the x-axis, `charges` on the y-axis, and with the color of each dot representing whether the subject is a smoker or not.

```r
library(ggplot2)

q1c_plot <- ggplot(df, aes(x = bmi, y = charges)) +
  geom_point(aes(color = smoker)) +
  labs(title = "Individual medical costs billed by health insurance vs. BMI",
       x = "BMI",
       y = "Charges",
       color = "Smoker status") +
  theme_bw() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major = element_blank())
q1c_plot
```
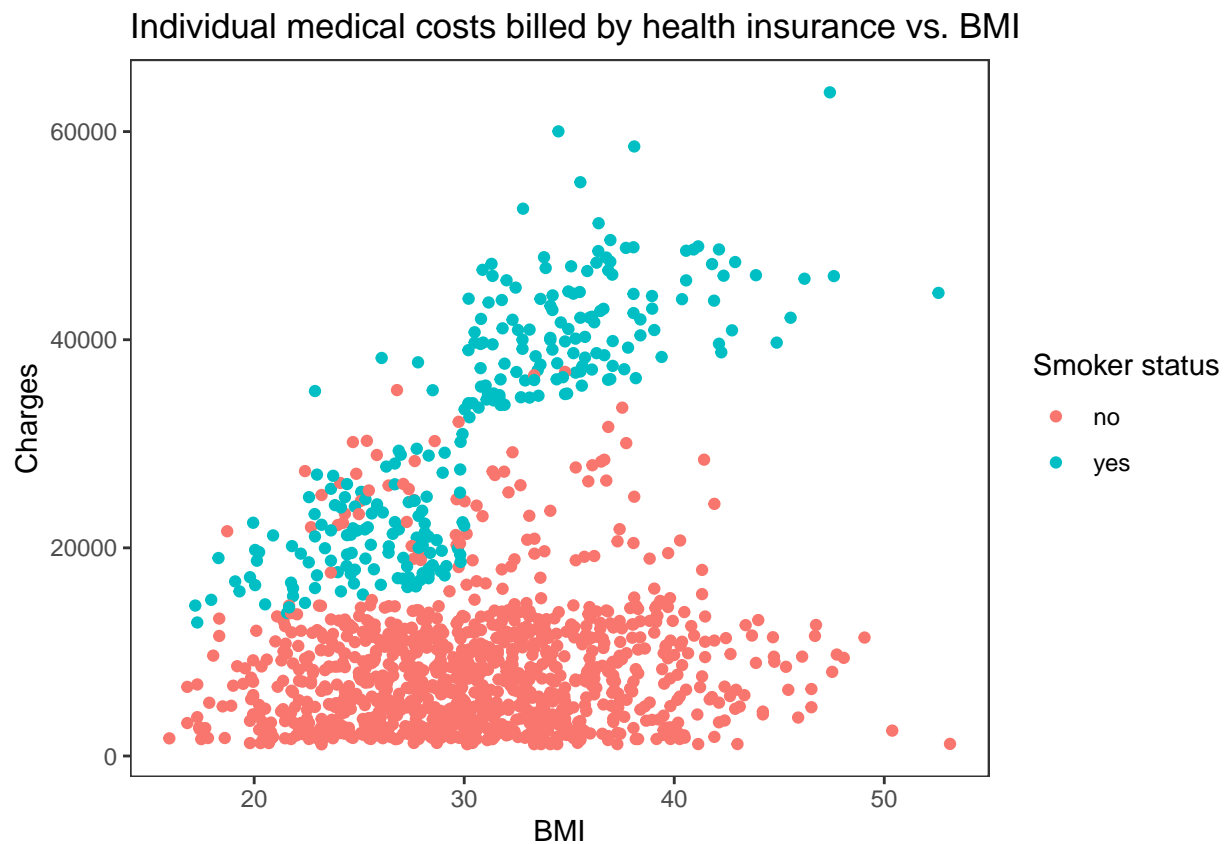

(d) Fit a least-squares linear model, with intercept, in order to predict:

- `charges` using `bmi` as the only predictor;
- `charges` using `bmi` and `smoker` as predictors;
- `charges` using `bmi` and `smoker` as in the previous model; but allowing for an interaction term between the variables `bmi` and `smoker`

For each of the three models:

- Present your results in the form of a table where you report the estimated regression coefficients and their interpretations (be careful with the dummy variables);

- Report the 95% confidence interval for the coefficient of the variable `bmi`, and provide a sentence explaining the meaning of this confidence interval;
- Draw the regression line(s) of the model on the scatter plot produced in point (c);
- Report the (training set) mean squared error of the model;
- Predict the medical costs billed by the health insurance company to a smoker with a `bmi` that is 29 and 31.5;
- Compute the predicted difference in charges between a smoker with `bmi` 31.5 and one with `bmi` 29. Do the same for non-smokers. Comment on the results

```r
results_table <- function(linear_model) {
  coeffs <- summary(linear_model)$coefficients
  row_length <- dim(coeffs)[1]
  CI_lower <- vector(mode = "numeric", length = row_length)
  CI_upper <- vector(mode = "numeric", length = row_length)
  cbind(coeffs, CI_lower, CI_upper)
}
```

```r
# linear model using `bmi` as the only predictor
charges_lm1 <- lm(charges ~ bmi, data = df)
```

```r
# linear model using `bmi` and `smoker` as predictors
charges_lm2 <- lm(charges ~ bmi + smoker, data = df)
```

```r
# linear model using `bmi` and `smoker` as predictors but also allowing for an
# interaction term between predictors `bmi` and `smoker`
charges_lm3 <- lm(charges ~ bmi + smoker + bmi*smoker , data = df)
```

(e) Now define and add to the dataset a new boolean variable `smoker_bmi30p` that is `True` only if the subject is a smoker **and** has a `bmi` greater than 30. Use this newly defined variable, together with `bmi` and `smoker`, to fit the linear model represented in Figure 1 by carefully defining the interaction terms (allow each of the three straight lines to have their own intercept and slope, but use the command `lm` only once).

- Present your results in the form of one table where you report the estimated coefficients of the model.
- For each predictor, comment on whether you can reject the null hypothesis that there is no (linear) association between that predictor and `charges`, conditional on the other predictors in the model.
- Explain the interpretation of the non-significant variables in the model ($p > 0.05$) and explain how Figure 1 would change if we were to discard those variables, i.e., perform variable selection.
- According to this newly defined model, compute the predicted difference in `charges` between a smoker with `bmi` 31.5 and one with `bmi` 29. Do the same for non-smokers. Compare the analogous results in point (d) and comment on the results.

Q2. This problem has to do with the notation of bias-variance trade-off. For (a) and (b), it's okay to submit hand-sketched plots: this is a conceptual exercise.

(a) Make a plot, like the one we saw in class, with "flexibility" on the x-axis. Sketch the following curves: squared bias, variance, irreducible error, expected prediction error. Be sure to label each curve. Indicate which level of flexibility is *best*.

(b) Make a plot with "flexibility" on the x-axis. Sketch curves corresponding to the training error and the test error. Be sure to label each curve. Indicate which level of flexibility is "best".

3

Q3. This problem has to do with numerical explorations of the bias-variance trade-off phenomenon. You will generate simulated data, and will use these data to perform **linear regression**. Set the seed with `set.seed(0)` before you begin.

(a) Use the `rnorm()` function to generate a predictor vector X of length $n = 30$, and use `runif()` to generate a noise vector ϵ of length $n = 30$.

(b) Generate a response vector Y of length $n = 30$ according to the model