STAT 435 HW1

Peiran Chen

3/31/2022

1.

a)

Taking a parametric approach will have following pros:
- Does not need a lot of data
- Simplifies the problem because it is generally much easier to estimate a set of parameters and cons:
- The model we choose will usually not match the true unknown form of f, and if the chosen model is too far from the true f, then our estimate will be poor.

Taking a nonparametric approach will have following pros:
- Avoid unnecessary assumptions about the functional form of f will have the potential to accurately fit a wider range of possible shapes for f.

and cons:
- A very large number of observations is required in order to obtain an accurate estimate for f.

b)

For parametric approach, I would say when we have a small number of observations to work with, such as getting survey on people's blood pressure and hours of physical exercises they do each week. And we know that having more time to exercises will result in a lower blood pressure as a matter of fact. Hence we can make assumptions to f in this case to be a linear model:

$$blood\,pressure \approx \beta_0 + \beta_1 \times physical\ exercises$$

c)

For non-parametric approach, I would say when we have a lot of data to work with, we can use this method to do the same prediction as part b).

2.

a)

In this case, I would expect the inflexible methods to perform better. Since sample size is small, there won't be enough data for flexible methods such as deep learning and etc. Also, the number of predictors are large, hence it would be a good practice to use OLS so that we have more interpretability. Hence inflexible methods tends to perform better.

b)

In this case, I would expect the flexible method to perform better.

3.

a)

It is a regression problem. And the goal is prediction, where $n = 50, p = 8$.

b)

It is a classification problem. And the goal is inference, where $n = 50, p = 6$.

4.

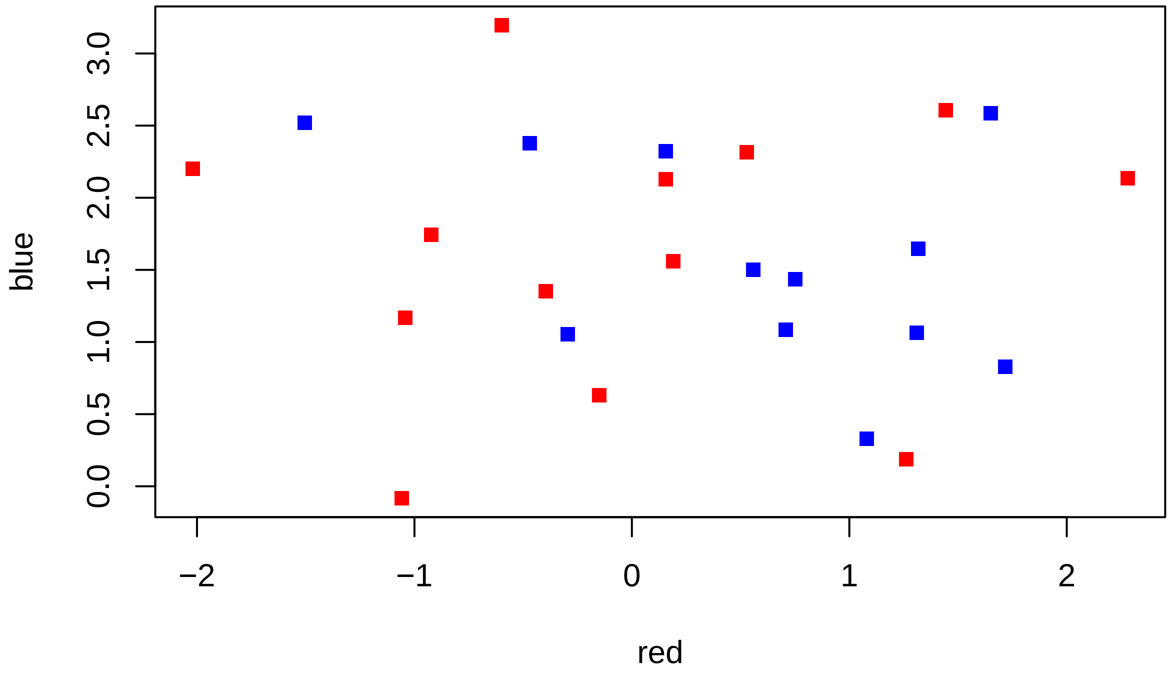
a)

5.

a)

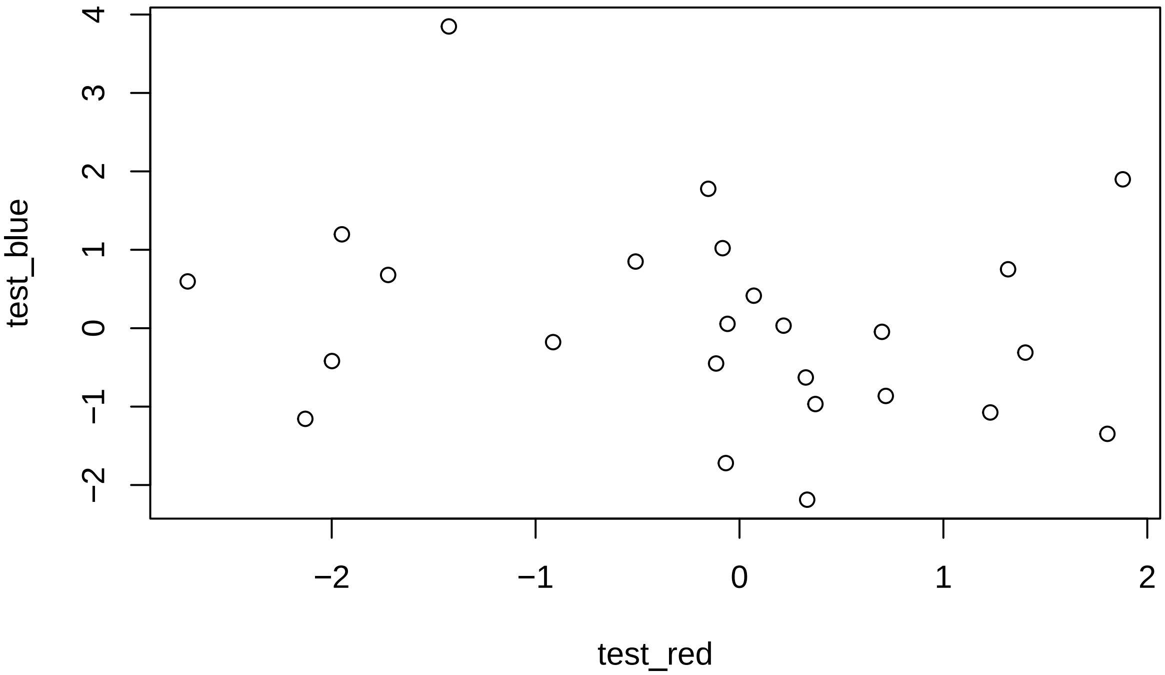
```r
n <- 25
red <- rnorm(n, 0, 1)
blue <- rnorm(n, 1.5, 1)
df_train <- data.frame("red" = red, "blue" = blue)

plot(df_train,
     pch = 15,
     col = c("red", "blue"))
```



```r
# Generating test set
test_red <- rnorm(n, 0, 1)
test_blue <- rnorm(n, 0, 1.5)
df_test <- data.frame("test_red" = test_red, "test_blue" = test_blue)

plot(df_test)
```



6.

a)

7.

```r
library(ISLR2)
```

a)

```r
data <- Boston
head(data)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

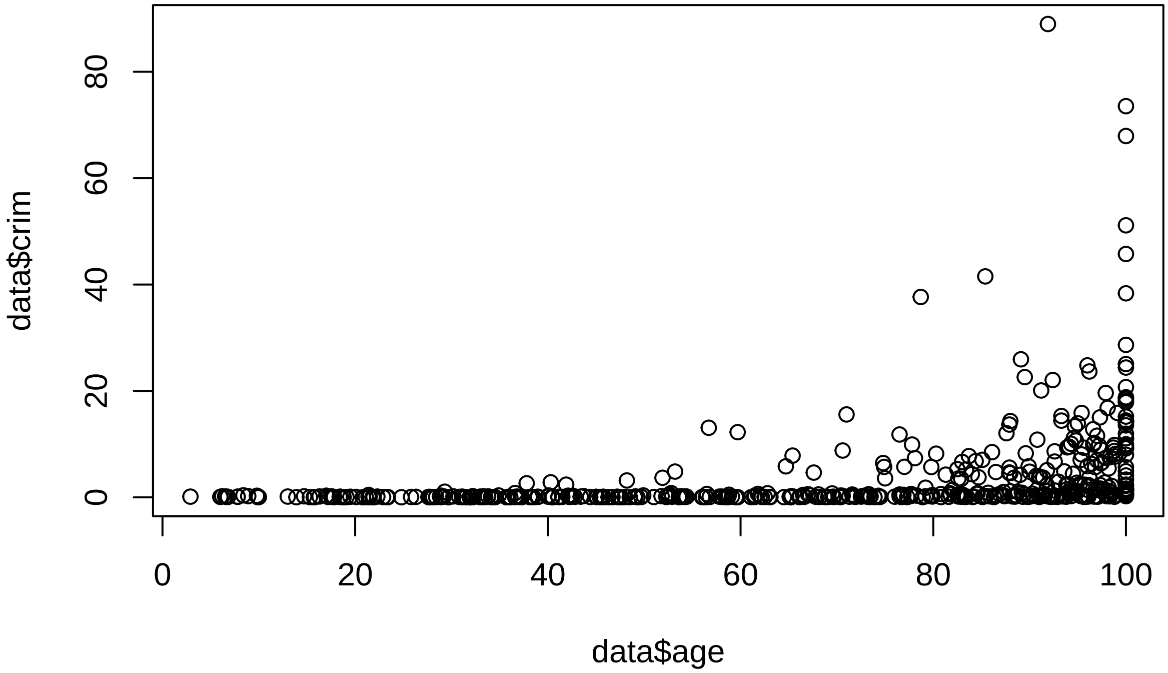
```
# Get number of rows
row_number <- nrow(data)

# Get number of columns
col_number <- ncol(data)
```

Number of rows represent the number of observations in our dataset. And number of columns represent the number of predictors we have.

b)

```
plot(data$age, data$crim)
```


data$crim
data$age