Homework 1

Rento Saijo
Department of Mathematics, Connecticut College
STA336: Statistical Machine Learning
Yan Zhuang, Ph.D.

January 30th, 2026

Problem 1

A very flexible approach has the advantage that it can represent a much wider range of possible shapes for f, and thus capture complicated (often non-linear) relationships between predictors X and a response Y. In contrast, a restrictive method like linear regression can only produce linear functions, e.g., $f(X) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$. The drawback of high flexibility is reduced interpretability: the fitted $\hat{f}$ can become so complex that it is difficult to understand how any individual predictor X_j is associated with Y, making flexible methods less attractive when inference and interpretability are the goal. Therefore, more flexible approaches are generally preferred when interpretability is not a priority and prediction is the primary objective, since we are willing to trade a clear description of predictor-response relationships for the ability to fit complex patterns; however, even for prediction, the most flexible model is not always best because highly flexible methods can overfit, so a less flexible method can sometimes yield better test performance. Conversely, a less flexible approach is preferred when inference is the goal because restrictive models are much more interpretable. *Source: ISLR2 §2.1.3, p. 24-6.*

Problem 2 (a)

```r
# Load libraries.
suppressMessages(library(tidyverse))
suppressMessages(library(GGally))
suppressMessages(library(ISLR2))

# Load data.
data(Auto)

# Count missing values.
colSums(is.na(Auto)) # It seems that we have no missing values.
```

```
##          mpg    cylinders displacement   horsepower       weight acceleration
##            0            0            0            0            0            0
##         year       origin         name
##            0            0            0
```

```r
# Check structure.
tibble::glimpse(Auto)
```

```
## Rows: 392
## Columns: 9
## $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 15, 14, 2~
## $ cylinders    <int> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, 4, ~
## $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390, 383, 34~
## $ horsepower   <int> 130, 165, 150, 150, 140, 198, 220, 215, 225, 190, 170, 16~
## $ weight       <int> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4425, 385~
## $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10.0, 8.5, ~
## $ year         <int> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 7~
## $ origin       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, ~
## $ name         <fct> chevrolet chevelle malibu, buick skylark 320, plymouth sa~
```

In the `Auto` data set, `mpg` is a quantitative variable but it is typically the response, not a predictor. The quantitative predictors are therefore `cylinders`, `displacement`, `horsepower`, `weight`, `acceleration`, and `year`. The qualitative predictors are `origin` (a categorical variable encoded numerically) and `name`.

Problem 2 (b)

```r
# Select quantitative predictors.
Auto_quant_preds <- Auto %>%
    dplyr::select(cylinders, displacement, horsepower, weight, acceleration, year)

# Compute range for each quantitative predictor.
ranges <- sapply(Auto_quant_preds, range)
tibble::tibble(
  Predictor = colnames(ranges),
  Min       = ranges[1, ],
  Max       = ranges[2, ],
  Range     = Max - Min
)
```

```
## # A tibble: 6 x 4
##   Predictor      Min   Max  Range
##   <chr>        <dbl> <dbl>  <dbl>
## 1 cylinders        3     8      5
```

```
## 2 displacement    68   455    387
## 3 horsepower      46   230    184
## 4 weight        1613 5140   3527
## 5 acceleration     8    24.8   16.8
## 6 year            70    82     12
```

```r
rm(ranges)
```

Problem 2 (c)

```r
# Compute mean and standard deviation for each.
tibble::tibble(
  Predictor = names(Auto_quant_preds),
  Mean      = sapply(Auto_quant_preds, mean),
  SD        = sapply(Auto_quant_preds, sd)
)
```

```
## # A tibble: 6 x 3
##   Predictor       Mean     SD
##   <chr>          <dbl>  <dbl>
## 1 cylinders       5.47   1.71
## 2 displacement  194.   105.
## 3 horsepower    104.    38.5
## 4 weight       2978.   849.
## 5 acceleration   15.5    2.76
## 6 year           76.0    3.68
```

Problem 2 (d)

```r
# Remove 10th through 85th observations (inclusive).
Auto_quant_subset <- Auto_quant_preds[-c(10:85), ]


# Compute range, mean, and standard deviation for each predictor on the subset.
ranges_sub <- sapply(Auto_quant_subset, range)
tibble::tibble(
```

```
  Predictor = colnames(ranges_sub),
  Min       = ranges_sub[1, ],
  Max       = ranges_sub[2, ],
  Range     = Max - Min,
  Mean      = sapply(Auto_quant_subset, mean),
  SD        = sapply(Auto_quant_subset, sd)
)
```

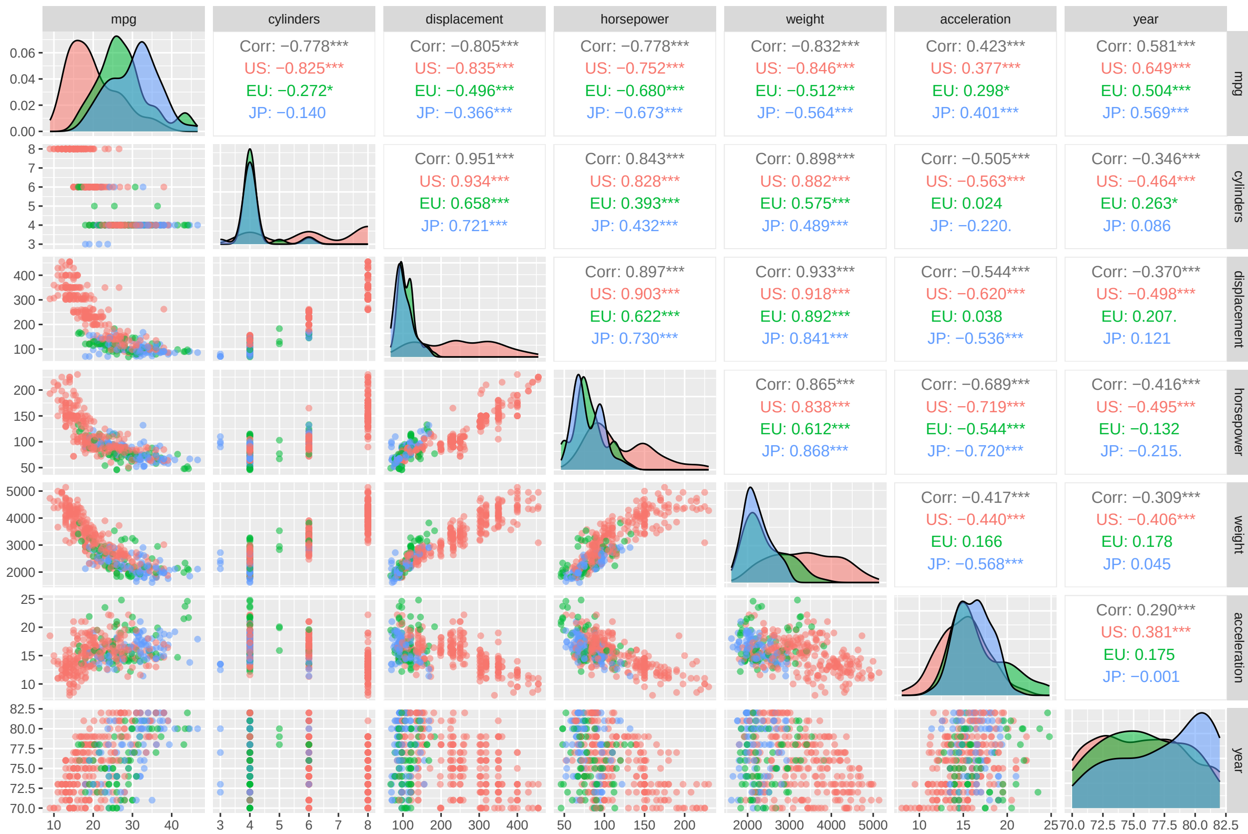
```
## # A tibble: 6 x 6
##   Predictor      Min   Max Range   Mean    SD
##   <chr>        <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1 cylinders      3     8     5     5.37  1.65
## 2 displacement  68   455   387   187.   99.7
## 3 horsepower    46   230   184   101.   35.7
## 4 weight      1649  4997  3348  2936.  811.
## 5 acceleration   8.5  24.8  16.3  15.7   2.69
## 6 year          70    82    12    77.1   3.11
```

```
rm(Auto_quant_preds, Auto_quant_subset, ranges_sub)
```

Problem 2 (e)

```
# Factor origin.
Auto_plot <- Auto %>%
  dplyr::mutate(origin = factor(origin, labels = c('US', 'EU', 'JP')))

# Inspect pairwise relationships.
GGally::ggpairs(
  Auto_plot,
  columns   = c('mpg', 'cylinders', 'displacement', 'horsepower', 'weight', 'acceleration', 'year'),
  aes(color = origin, alpha = 0.5)
)
```



The scatterplot matrix shows strong collinearity among the "size/power" predictors: `cylinders`, `displacement`, `horsepower`, and `weight` move together very tightly (e.g., `cylinders-displacement` has a correlation around 0.95, and `displacement-weight` around 0.93), suggesting these variables are largely measuring the same underlying concept (bigger engines/cars tend to be heavier and more powerful). In contrast, `acceleration` tends to be negatively associated with those size/power variables (most notably with `horsepower`, around -0.69, and with `displacement`, around -0.54), indicating that cars with larger engines and greater power/weight tend to have smaller acceleration values in this dataset. The variable `year` is moderately negatively related to the size/power measures (roughly -0.31 to -0.42 with `weight`, `displacement`, and `horsepower`) and mildly positively related to `acceleration` (about 0.29), consistent with cars becoming lighter and less "big-engine" over time. Finally, the color-group patterns by origin suggest systematic differences across regions (U.S. cars clustering at higher weight/displacement/horsepower), and the within-origin correlations sometimes differ (e.g., the `cylinders-acceleration` relationship is much stronger for U.S. cars than for European or Japanese cars), reinforcing that relationships among predictors can vary by subgroup even when the overall trend is clear.