# Homework 1

Rento Saijo

Department of Mathematics, Connecticut College

STA336: Statistical Machine Learning

Yan Zhuang, Ph.D.

January 30th, 2026

### AI Disclosure Statement

ChatGPT-5.2 (Thinking) was used to create the YAML portion to format the text nicely; I looked into the documentation for each of the package it used and added/removed unnecessary formattings.

### 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)

```
# 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
```

```
# 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, 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)**

```
# 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
```

```
rm(ranges)
```

**Problem 2 (c)**

```
# 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)**

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

```r
# 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
```

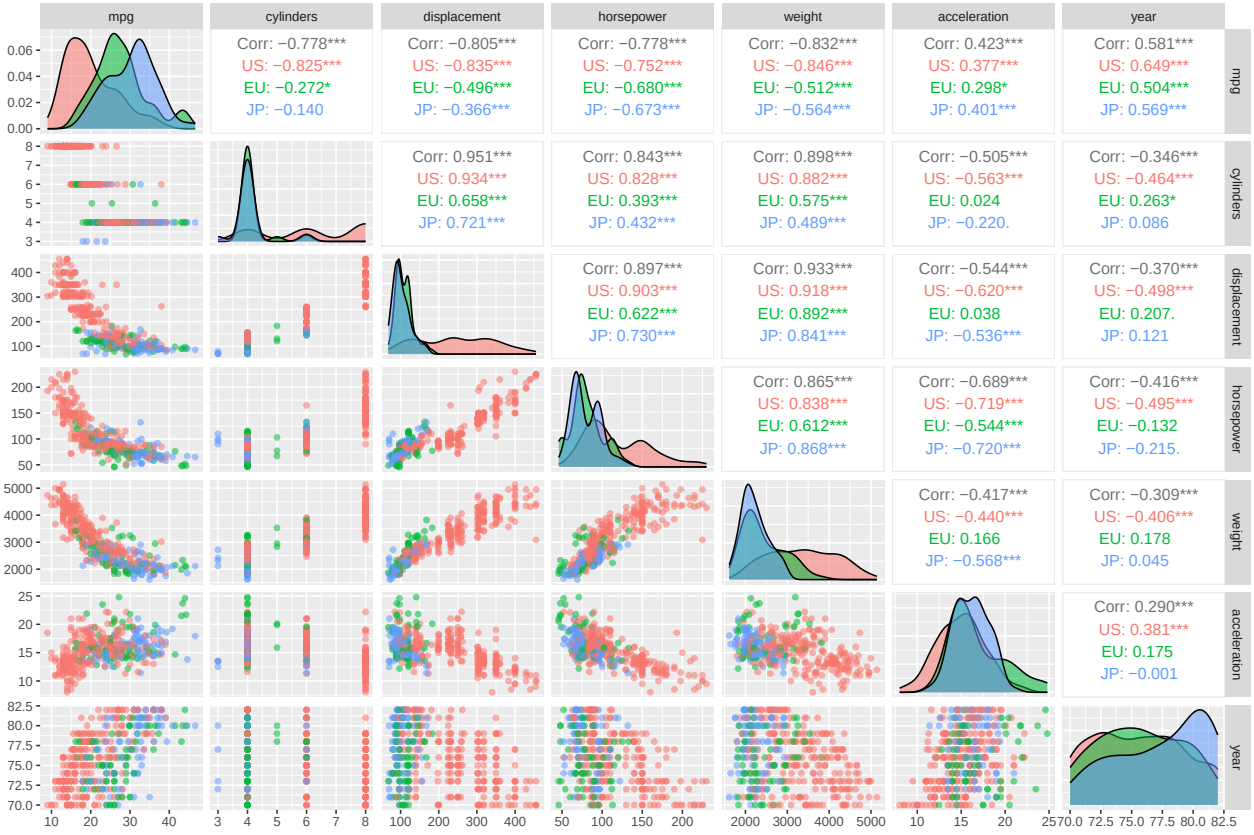
```r
rm(Auto_quant_preds, Auto_quant_subset, ranges_sub)
```

**Problem 2 (e)**

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


# Inspect pairwise relationships.
GGally::ggpairs(
  Auto_plot,
  columns   = setdiff(colnames(Auto_plot), c('origin', 'name')),
```

```
  aes(color = origin, alpha = 0.5)
)
```




```
rm(Auto_plot)
```

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.

### Problem 2 (f)

Yes. The plots suggest that several variables should be useful for predicting `mpg`. In particular, `mpg` has strong negative associations with `weight`, `displacement`, `horsepower`, and `cylinders` (heavier, larger-engine, higher-power cars tend to have lower gas mileage), so these predictors should have substantial predictive value. The plots also show that `mpg` increases with `year`, indicating that newer model years are generally more fuel-efficient, and `mpg` has a more moderate positive relationship with `acceleration`. Finally, the clear separation by `origin` in the panels suggests that `origin` is also informative: cars from different regions cluster at different typical fuel-efficiency levels even after accounting for other predictors, so it may help explain additional variation in `mpg` beyond the purely quantitative variables.

### Problem 3 (a)

```
# Load data.
data(Boston)


# Learn data.
# ?Boston


# Check dimensions.
dim(Boston)
```

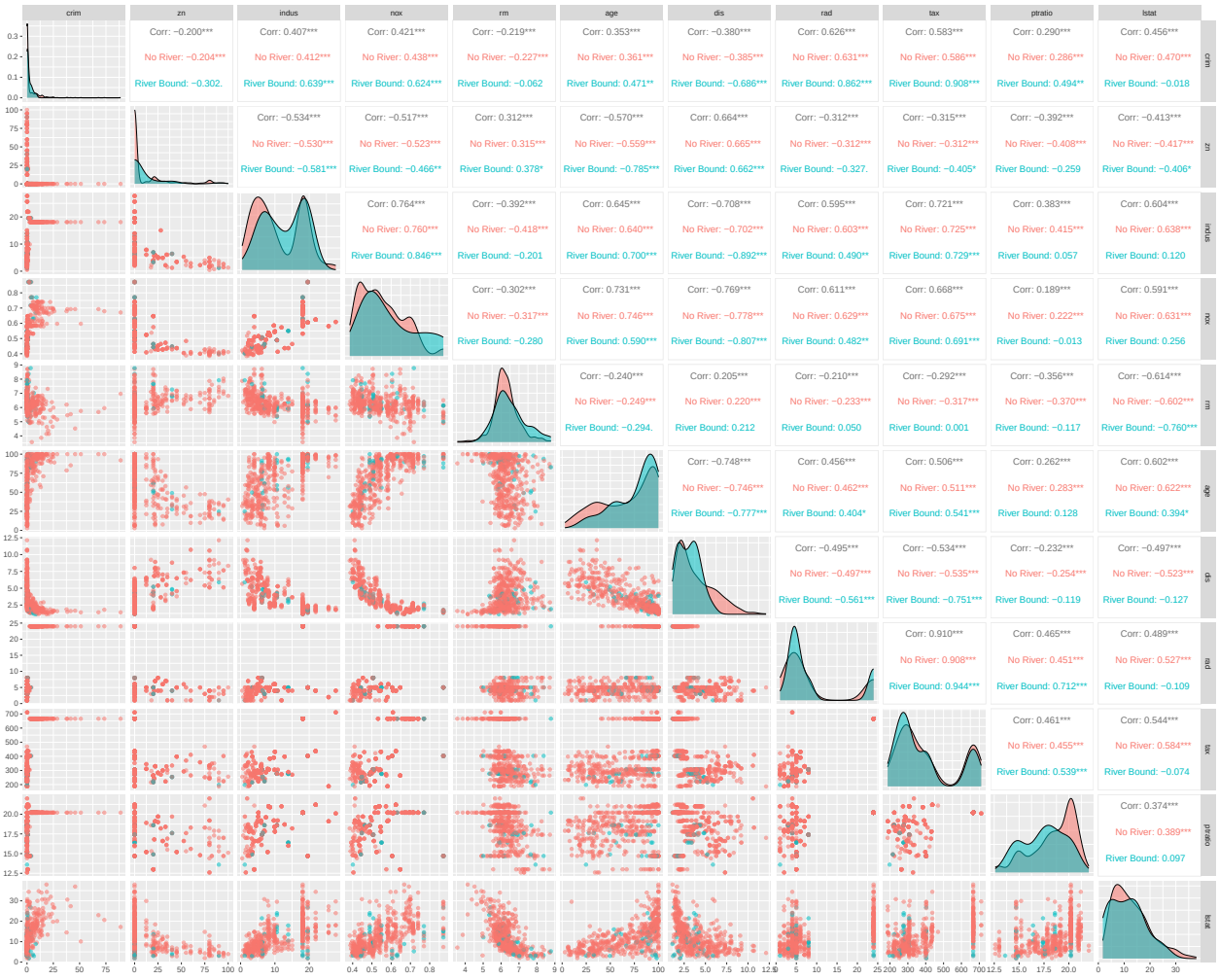
```
## [1] 506  13
```

There are 506 rows and 13 columns in the `Boston` data set. Each row corresponds to a census tract in the Boston area, and each column is a variable recorded for that tract. One of the columns is `medv`, the median home value (in $1000s), and the remaining columns are predictors describing characteristics of the tract (e.g., crime rate, number of rooms, property tax rate, etc.).

### Problem 3 (b)

```
# Factor origin.
Boston_plot <- Boston %>%
  dplyr::mutate(chas = factor(chas, levels = c(0, 1), labels = c('No River', 'River Bound')))
```

```
# Inspect pairwise relationships.
GGally::ggpairs(
  Boston_plot,
  columns = setdiff(colnames(Boston_plot), c('medv', 'chas')),
  aes(color = chas, alpha = 0.5)
)
```




```
rm(Boston_plot)
```

Many predictors are highly skewed, especially `crim`, `zn`, `rad`, and `tax`, with most observations near small values and a handful of extreme outliers, so relationships are often driven by a small number of high-leverage tracts. There is clear collinearity among "urban/industrial" variables: `indus` is strongly positively associated with `nox`, while `dis` is strongly negatively associated with both `indus` and `nox`, suggesting that more industrial and higher-pollution tracts tend to lie closer to employment centers. The variable `age`

also tends to be higher where `dis` is lower, consistent with older housing stock being concentrated nearer the city. In addition, `rad` and `tax` are very strongly positively related, indicating that tracts with greater accessibility to radial highways tend to have higher property-tax rates. Finally, `lstat` is positively related to several "urban stress" measures (e.g., `crim`, `nox`, `tax`) and negatively related to variables associated with more desirable suburban tracts (e.g., `zn`, `dis`), reinforcing that multiple predictors are capturing overlapping aspects of neighborhood socioeconomic status and urbanization. The binary predictor `chas` (river adjacency) naturally appears as two bands; any differences involving `chas` are best interpreted as group-level shifts rather than a continuous linear trend.

### Problem 3 (c)

Yes, several predictors appear associated with the per-capita crime rate `crim`, and the relationships are fairly systematic in the scatterplot matrix. `crim` tends to be higher in tracts with more industrial/urban characteristics: it increases with `rad` and `tax` (areas with greater highway accessibility and higher tax rates), and it is also positively associated with `indus`, `nox`, `ptratio`, and `lstat` (more non-retail business activity, higher pollution, higher pupil–teacher ratios, and a higher percent "lower status" population). In contrast, `crim` tends to be lower in tracts that look more suburban or higher-value: it decreases as `dis` increases (farther from employment centers), and it is negatively related to `zn`, and `rm` (more large-lot zoning and more rooms per dwelling).

### Problem 3 (d)

```
# List tracts with highest crime rates.
Boston_id <- Boston %>%
  dplyr::mutate(tract = dplyr::row_number())
Boston_id %>%
    dplyr::arrange(dplyr::desc(crim)) %>%
    dplyr::slice(1:5) %>%
    dplyr::select(tract, crim, tax, ptratio, medv)
```

```
##   tract    crim tax ptratio medv
## 1   381 88.9762 666    20.2 10.4
## 2   419 73.5341 666    20.2  8.8
## 3   406 67.9208 666    20.2  5.0
## 4   411 51.1358 666    20.2 15.0
```

```
## 5   415 45.7461 666    20.2  7.0
```

```
# List tracts with highest tax rates.
Boston_id %>%
    dplyr::arrange(dplyr::desc(tax)) %>%
    dplyr::slice(1:5) %>%
    dplyr::select(tract, tax, crim, ptratio, medv)
```

```
##   tract tax    crim ptratio medv
## 1   489 711 0.15086    20.1 15.2
## 2   490 711 0.18337    20.1  7.0
## 3   491 711 0.20746    20.1  8.1
## 4   492 711 0.10574    20.1 13.6
## 5   493 711 0.11132    20.1 20.1
```

```
# List tracts with highest pupil-teacher ratios.
Boston_id %>%
    dplyr::arrange(dplyr::desc(ptratio)) %>%
    dplyr::slice(1:5) %>%
    dplyr::select(tract, ptratio, crim, tax, medv)
```

```
##   tract ptratio    crim tax medv
## 1   355    22.0 0.04301 334 18.2
## 2   356    22.0 0.10659 334 20.6
## 3   128    21.2 0.25915 437 16.2
## 4   129    21.2 0.32543 437 18.0
## 5   130    21.2 0.88125 437 14.3
```

```
# Compute ranges.
ranges <- sapply(Boston %>% dplyr::select(crim, tax, ptratio), range)
tibble::tibble(
  Variable = colnames(ranges),
  Min      = ranges[1, ],
  Max      = ranges[2, ],
  Range    = Max - Min
)
```

```
## # A tibble: 3 x 4
##   Variable       Min   Max Range
##   <chr>        <dbl> <dbl> <dbl>
## 1 crim       0.00632  89.0  89.0
## 2 tax      187       711   524
## 3 ptratio   12.6      22     9.4
```

```
rm(ranges)
```

Yes, there are several census tracts that stand out as having particularly high values of the requested predictors. For crime rate, a small set of tracts have extremely large `crim` values (up to about 89), far above the bulk of observations (minimum about 0.006), indicating clear outliers and a very wide range (roughly 89). For tax rates, there are tracts with `tax` as high as 711 (minimum 187), giving a large range of 524, and the highest-tax tracts are clearly separated from most towns. For pupil–teacher ratio, the maximum is 22.0 and the minimum is 12.6, so the range (9.4) is much narrower than for `crim` or `tax`; although some tracts do have notably high `ptratio`, the spread is comparatively modest. Overall, the ranges show that `crim` and `tax` vary dramatically across tracts (with especially extreme high values), while `ptratio` varies less.

**Problem 3 (e)**

```
# Count tract(s) bounding Charles River.
sum(Boston$chas == 1)
```

```
## [1] 35
```

**Problem 3 (f)**

```
# Compute median pupil-teacher ratio.
median(Boston$ptratio, na.rm = TRUE)
```

```
## [1] 19.05
```

**Problem 3 (g)**

```
# Identify tract(s) with lowest medv.
min_medv_tracts <- Boston_id %>% dplyr::slice_min(medv, n = 1, with_ties = TRUE)
min_medv_tracts
```

```
##      crim zn indus chas   nox    rm age    dis rad tax ptratio lstat medv tract
## 1 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 30.59    5   399
## 2 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 22.98    5   406
```

```
# Define helper to make comparison tables.
make_comp_tbl <- function(tid, Boston_id, pred_names, ranges) {
    row <- Boston_id %>% dplyr::filter(tract == tid)
    v   <- as.numeric(row[1, pred_names])
    tibble::tibble(
        Predictor = pred_names,
        Value     = v,
        Min       = ranges[1, ],
        Max       = ranges[2, ],
        Range     = Max - Min
    )
}

# Compare values.
tract_ids  <- min_medv_tracts$tract
pred_names <- setdiff(names(Boston), 'medv')
ranges     <- sapply(Boston %>% dplyr::select(dplyr::all_of(pred_names)), range)
make_comp_tbl(399, Boston_id, pred_names, ranges)
```

```
## # A tibble: 12 x 5
##    Predictor   Value       Min     Max   Range
##    <chr>       <dbl>     <dbl>   <dbl>   <dbl>
##  1 crim       38.4     0.00632  89.0    89.0
##  2 zn          0       0       100     100
##  3 indus      18.1     0.46     27.7    27.3
##  4 chas        0       0         1       1
##  5 nox         0.693   0.385     0.871   0.486
##  6 rm          5.45    3.56      8.78    5.22
##  7 age       100       2.9     100      97.1
##  8 dis         1.49    1.13     12.1    11.0
```

```
##  9 rad        24       1        24      23
## 10 tax       666     187       711     524
## 11 ptratio    20.2    12.6      22       9.4
## 12 lstat      30.6     1.73     38.0    36.2
```

```
make_comp_tbl(406, Boston_id, pred_names, ranges)
```

```
## # A tibble: 12 x 5
##    Predictor   Value       Min     Max   Range
##    <chr>       <dbl>     <dbl>   <dbl>   <dbl>
##  1 crim       67.9     0.00632  89.0    89.0
##  2 zn          0       0       100     100
##  3 indus      18.1     0.46     27.7    27.3
##  4 chas        0       0         1       1
##  5 nox         0.693   0.385     0.871   0.486
##  6 rm          5.68    3.56      8.78    5.22
##  7 age       100       2.9     100      97.1
##  8 dis         1.43    1.13     12.1    11.0
##  9 rad        24       1        24      23
## 10 tax       666     187       711     524
## 11 ptratio    20.2    12.6      22       9.4
## 12 lstat      23.0     1.73     38.0    36.2
```

```
rm(min_medv_tracts, make_comp_tbl, tract_ids, pred_names, ranges)
```

The lowest median home value `medv` occurs in two census tracts, 399 and 406, both with `medv = 5` (in $1000s). For tract 399, several predictors are at or near extreme values relative to their overall ranges: `age = 100` equals the dataset maximum, `rad = 24` equals the maximum, `tax = 666` is near the upper end of the interval $[187, 711]$, `nox = 0.693` is high relative to $[0.385, 0.871]$, and `dis  1.49` is close to the minimum within $[1.13, 12.1]$. It also has unfavorable socioeconomic/urban indicators: `lstat  30.6` is high within $[1.73, 38.0]$, and `crim  38.4` is extremely high within $[0.00632, 89.0]$. Tract 406 shows a very similar pattern: `age = 100`, `rad = 24`, `tax = 666`, `nox = 0.693`, and `dis  1.43` are again near the extremes, while its `crim  67.9` is even higher than tract 399's, and `lstat  23.0` is still elevated within $[1.73, 38.0]$. Overall, these two tracts combine high crime, high taxes, high pollution/industrial indicators, very old housing, and proximity to the urban core (low `dis`), which is consistent with them having the lowest median home values in the

data.

### Problem 3 (h)

```
# Count tracts with rm > 7 and rm > 8.
tibble::tibble(
  Condition = c('rm > 7', 'rm > 8'),
  Count     = c(sum(Boston$rm > 7), sum(Boston$rm > 8))
)
```

```
## # A tibble: 2 x 2
##   Condition Count
##   <chr>     <int>
## 1 rm > 7       64
## 2 rm > 8       13
```

```
# Inspects tracts with rm > 8.
Boston_id %>%
  dplyr::relocate(tract, .before = 1) %>%
  dplyr::filter(rm > 8) %>%
  dplyr::arrange(dplyr::desc(rm))
```

```
##    tract    crim zn indus chas    nox    rm  age    dis rad tax ptratio lstat
## 1    365 3.47428  0 18.10    1 0.7180 8.780 82.9 1.9047  24 666    20.2  5.29
## 2    226 0.52693  0  6.20    0 0.5040 8.725 83.0 2.8944   8 307    17.4  4.63
## 3    258 0.61154 20  3.97    0 0.6470 8.704 86.9 1.8010   5 264    13.0  5.12
## 4    263 0.52014 20  3.97    0 0.6470 8.398 91.5 2.2885   5 264    13.0  5.91
## 5    164 1.51902  0 19.58    1 0.6050 8.375 93.9 2.1620   5 403    14.7  3.32
## 6    233 0.57529  0  6.20    0 0.5070 8.337 73.3 3.8384   8 307    17.4  2.47
## 7    268 0.57834 20  3.97    0 0.5750 8.297 67.0 2.4216   5 264    13.0  7.44
## 8    225 0.31533  0  6.20    0 0.5040 8.266 78.3 2.8944   8 307    17.4  4.14
## 9    254 0.36894 22  5.86    0 0.4310 8.259  8.4 8.9067   7 330    19.1  3.54
## 10   234 0.33147  0  6.20    0 0.5070 8.247 70.4 3.6519   8 307    17.4  3.95
## 11    98 0.12083  0  2.89    0 0.4450 8.069 76.0 3.4952   2 276    18.0  4.21
## 12   227 0.38214  0  6.20    0 0.5040 8.040 86.5 3.2157   8 307    17.4  3.13
```

```
## 13   205 0.02009 95  2.68    0 0.4161 8.034 31.9 5.1180   4 224    14.7  2.88
##    medv
## 1  21.9
## 2  50.0
## 3  50.0
## 4  48.8
## 5  50.0
## 6  41.7
## 7  50.0
## 8  44.8
## 9  42.8
## 10 48.3
## 11 38.7
## 12 37.6
## 13 50.0
```

```
rm(Boston_id)
```

In this data set, 64 census tracts have an average of more than 7 rooms per dwelling (`rm > 7`), and 13 tracts have an average of more than 8 rooms per dwelling (`rm > 8`). The `rm > 8` tracts are relatively rare and generally look like higher-end neighborhoods comparing with the summary of each predictors we saw in the previous problem part: many have very high median home values (`medv` is often near the top of the scale, including several at 50), low `lstat` values (roughly in the 2-7 range here), and typically low-to-moderate crime rates compared with the extreme-crime tracts seen elsewhere in the data. One notable exception is tract 365, which has `rm = 8.78` but also relatively high `crim` and high `nox`, showing that a tract can have very large houses while still having some urban disamenities. Overall, tracts with more than eight rooms per dwelling tend to correspond to especially desirable areas with high home values and favorable socioeconomic indicators.