Exercises Chapter 2

Contents

Conceptual 2

Applied 4

```r
knitr::opts_chunk$set(
  collapse = TRUE,
  comment = "#>"
)

library(MASS)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --

## √ ggplot2 2.2.1     √ purrr   0.2.4
## √ tibble  1.4.1     √ dplyr   0.7.4
## √ tidyr   0.7.2     √ stringr 1.2.0
## √ readr   1.1.1     √ forcats 0.2.0

## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```r
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(ISLR)
library(skimr)
```

```
##
## Attaching package: 'skimr'

## The following objects are masked from 'package:dplyr':
##
##     contains, ends_with, everything, matches, num_range, one_of,
##     starts_with
```

```r
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```r
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```r
theme_set(theme_minimal())
```

Conceptual

(1)

(a) Flexible methods should outperform inflexible methods because the large sample size prevents flexible methods from overfitting.
(b) When dealing with small samples and many predictors, flexible methods tend to overfit because they show higher variances. Therefore one should expect less flexible methods to perform better.
(c) Flexible methods will outperform inflexible methods since they are generally less biased, especially if the true relationship between predictors and response is non-linear.
(d) One might expect less flexible methods to perform better in this setting since they do not catch every bit of variance in the data and therefore provide more smoothing. Flexible methods on the other hand are likely to overfit.

(2)

(a) regression problem; inference; n = 500; p = 3
(b) classification problem; prediction; n = 20; p = 13
(c) regression problem; prediction; n = 52; p = 3

(3)

(a)

(b) The *bias curve* decreases monotonically since more flexible methods capture more of the variation in the data, resulting in lower bias. The *variance* curve increases monotonically because higher flexibility allows to reflect smaller details in the data. This results in higher variance. The *training error* decreases monotonically because more flexible methods can ultimately catch up every variation in the data, including white noise. Therefore the training error can be reduced to zero. The *test error* curve follows a U-shape. In the beginning, more flexibility leads to lower bias and therefore lower test errors. With increasing flexibility, methods begin to overfit the data by capturing white noise. The *Bayes error* curve is a horizontal line because

(4) To be added

(5) A very flexible approach is able to take into account very small bits of variation in the data. This makes flexible approaches prone to overfitting. If the true relation of of response and predictors is non-linear, more flexible approaches have advantages because they can reflect the non-linear relation better. Also, if prediction accuracy is more important than interpretability of a model, more flexible approaches might be better. Less flexible approaches have advantages when it comes to interpretability of results rather than prediction accuracy.

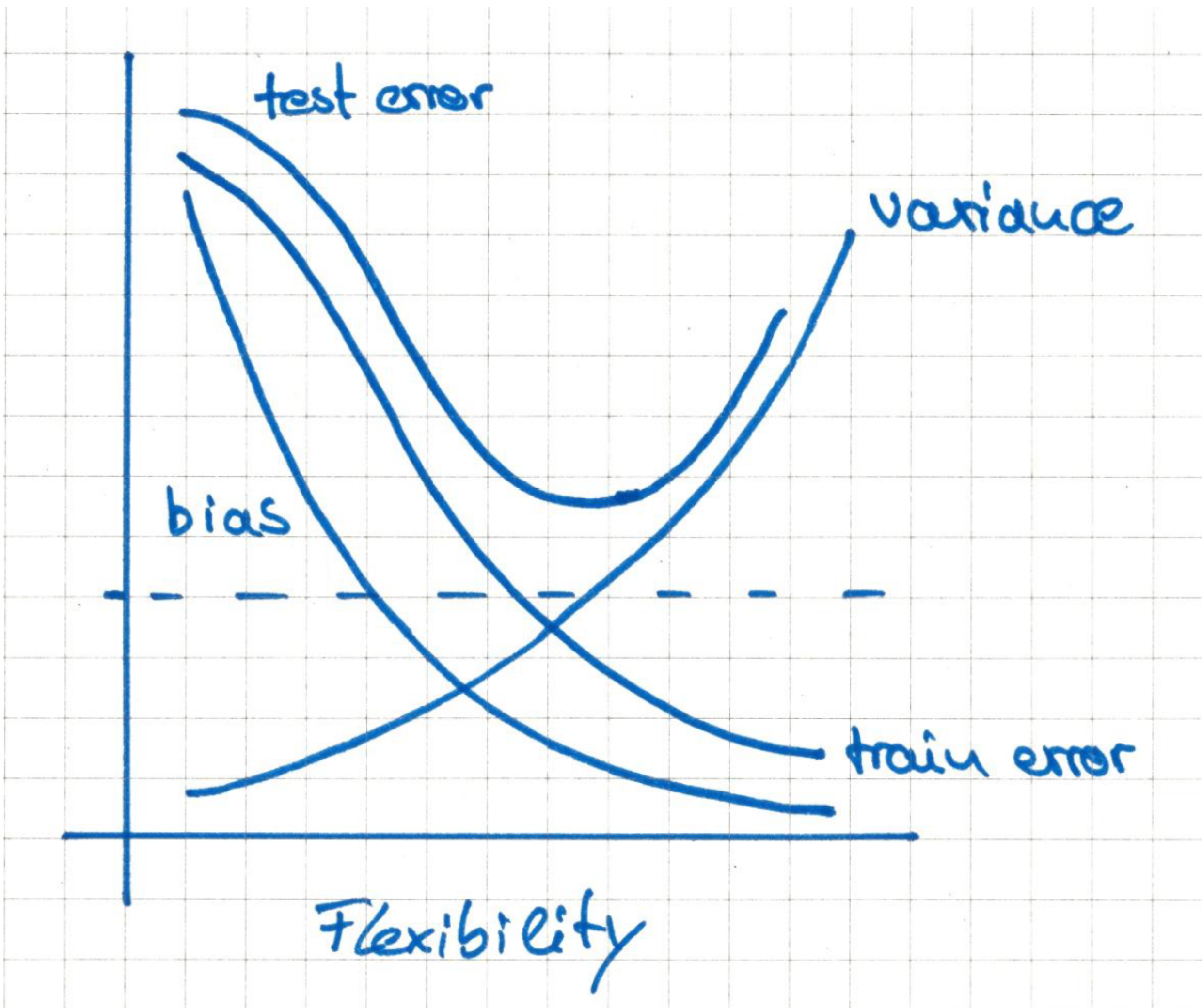


Figure 1:

(6) A parametric approach assumes a specific relation between response and predictors. This is good, when the true relationship is similar to the assumed one because fitting parametric models is easy in comparison to fitting non-parametric models. This is bad, if the assumed relationship is very different from the true relationship, resulting in a poor fit.

(7)

(a)

```
table <-
  tibble(obs = 1:6,
         X_1 = c(0, 2, 0, 0,-1, 1),
         X_2 = c(3, 0, 1, 1, 0, 1),
         X_3 = c(0, 0, 3, 2, 1, 1),
         Y = c("Red", "Red", "Red", "Green", "Green", "Red")) %>%
  mutate(eucl_dist = sqrt((X_1 - 0)^2 + (X_2 - 0)^2 + (X_3 - 0)^2))

table
#> # A tibble: 6 x 6
#>     obs   X_1   X_2   X_3 Y     eucl_dist
#>   <int> <dbl> <dbl> <dbl> <chr>     <dbl>
#> 1     1  0     3.00  0    Red        3.00
#> 2     2  2.00  0     0    Red        2.00
#> 3     3  0     1.00  3.00 Red        3.16
#> 4     4  0     1.00  2.00 Green      2.24
#> 5     5 -1.00  0     1.00 Green      1.41
#> 6     6  1.00  1.00  1.00 Red        1.73
```

(b)

```
table %>%
  arrange(eucl_dist)
#> # A tibble: 6 x 6
#>     obs   X_1   X_2   X_3 Y     eucl_dist
#>   <int> <dbl> <dbl> <dbl> <chr>     <dbl>
#> 1     5 -1.00  0     1.00 Green      1.41
#> 2     6  1.00  1.00  1.00 Red        1.73
#> 3     2  2.00  0     0    Red        2.00
#> 4     4  0     1.00  2.00 Green      2.24
#> 5     1  0     3.00  0    Red        3.00
#> 6     3  0     1.00  3.00 Red        3.16
```

My prediction is “Green” because obs 5 shows the lowest euclidian distance and Y(obs = 5) = “Green”.

(c) My prediction is “Red” because 2 out of those 3 obs with lowest euclidian distance have Y = “Red”.

(d) We would expect that the best value of K is rather low, because this allows for more variance in the predictions.

Applied

(8)

(a)

```
College <- as_tibble(College)
```

(b)

```
College %>%
  rownames_to_column(var = "University")
#> # A tibble: 777 x 19
#>    Univers~ Priva~  Apps Accept Enro~ Top1~ Top2~ F.Un~ P.Und~ Outs~ Room~
#>    <chr>    <fctr> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
#>  1 Abilene~ Yes     1660   1232 721    23.0  52.0  2885  537    7440  3300
#>  2 Adelphi~ Yes     2186   1924 512    16.0  29.0  2683 1227   12280  6450
#>  3 Adrian ~ Yes     1428   1097 336    22.0  50.0  1036   99.0 11250  3750
#>  4 Agnes S~ Yes      417    349 137    60.0  89.0   510   63.0 12960  5450
#>  5 Alaska ~ Yes      193    146  55.0  16.0  44.0   249  869    7560  4120
#>  6 Alberts~ Yes      587    479 158    38.0  62.0   678   41.0 13500  3335
#>  7 Albertu~ Yes      353    340 103    17.0  45.0   416  230   13290  5720
#>  8 Albion ~ Yes     1899   1720 489    37.0  68.0  1594   32.0 13868  4826
#>  9 Albrigh~ Yes     1038    839 227    30.0  63.0   973  306   15595  4400
#> 10 Alderso~ Yes      582    498 172    21.0  44.0   799   78.0 10468  3380
#> # ... with 767 more rows, and 8 more variables: Books <dbl>,
#> #   Personal <dbl>, PhD <dbl>, Terminal <dbl>, S.F.Ratio <dbl>,
#> #   perc.alumni <dbl>, Expend <dbl>, Grad.Rate <dbl>
```

(c)

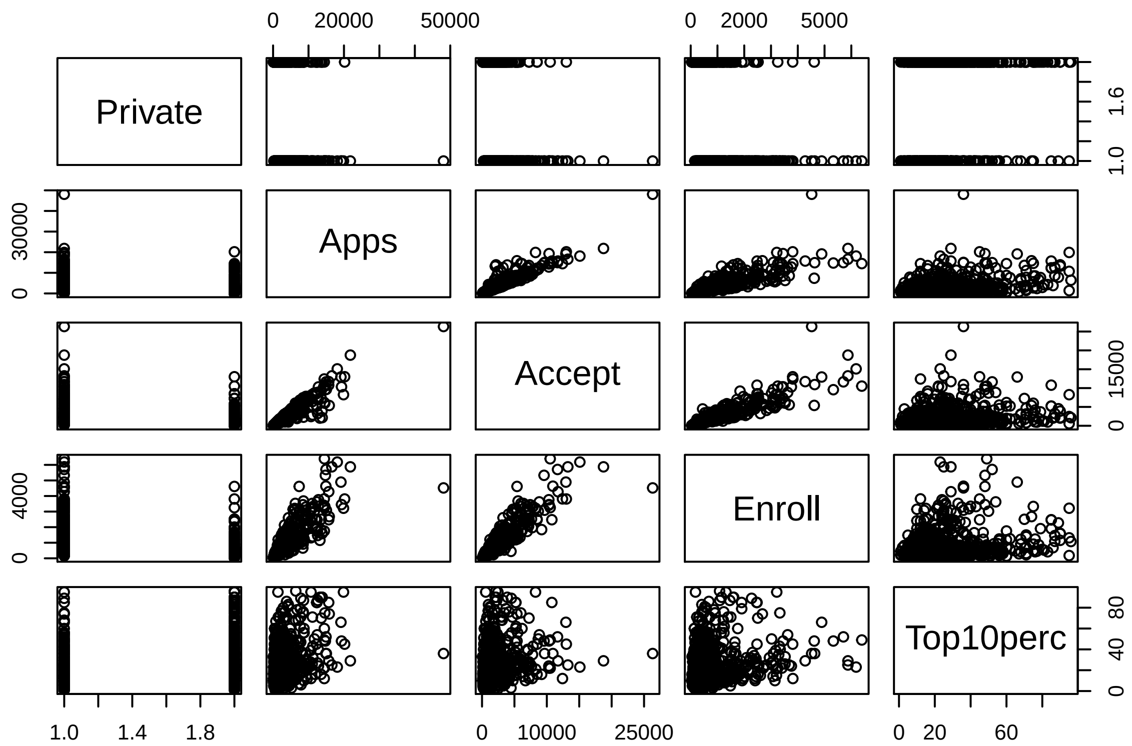
i.

```
summary(College)
#>  Private        Apps           Accept          Enroll       Top10perc
#>  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
#>  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
#>            Median : 1558   Median : 1110   Median : 434   Median :23.00
#>            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
#>            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
#>            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
#>    Top25perc      F.Undergrad     P.Undergrad         Outstate
#>  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
#>  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
#>  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
#>  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
#>  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
#>  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
#>    Room.Board       Books           Personal        PhD
#>  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
#>  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
#>  Median :4200   Median : 500.0   Median :1200   Median : 75.00
#>  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
#>  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
#>  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
#>     Terminal       S.F.Ratio      perc.alumni        Expend
#>  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
#>  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
#>  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
#>  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
#>  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
#>  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
#>    Grad.Rate
#>  Min.   : 10.00
```

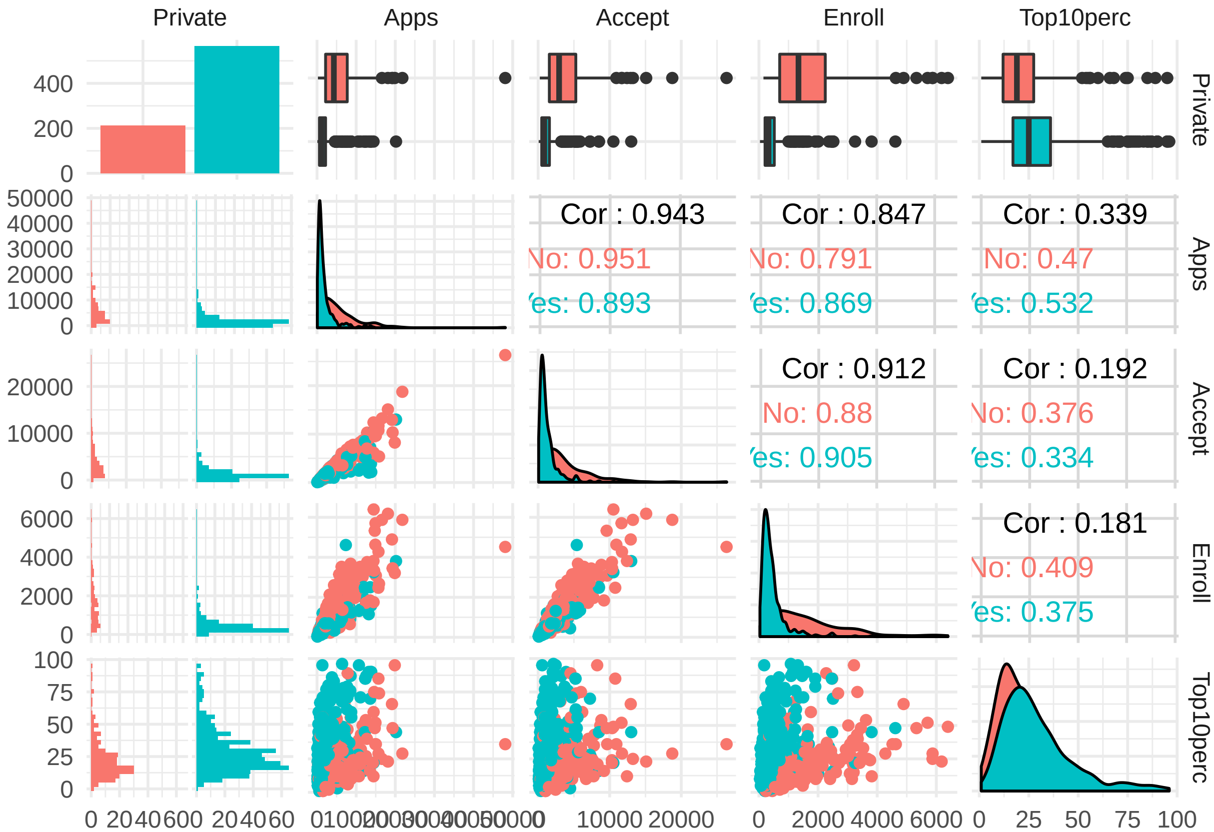
```
#>  1st Qu.: 53.00
#>  Median : 65.00
#>  Mean   : 65.46
#>  3rd Qu.: 78.00
#>  Max.   :118.00

skim(College)
#> Skim summary statistics
#>  n obs: 777
#>  n variables: 18
#>
#> Variable type: factor
#>   variable missing complete   n n_unique                top_counts ordered
#> 1  Private       0      777 777        2 Yes: 565, No: 212, NA: 0   FALSE
#>
#> Variable type: numeric
#>       variable missing complete   n     mean      sd    min    p25 median
#> 1       Accept       0      777 777   2018.8  2451.11   72    604   1110
#> 2         Apps       0      777 777  3001.64 3870.2    81    776   1558
#> 3        Books       0      777 777   549.38  165.11   96    470    500
#> 4       Enroll       0      777 777   779.97  929.18   35    242    434
#> 5       Expend       0      777 777  9660.17 5221.77 3186   6751   8377
#> 6  F.Undergrad       0      777 777  3699.91 4850.42  139    992   1707
#> 7    Grad.Rate       0      777 777    65.46   17.18   10     53     65
#> 8     Outstate       0      777 777 10440.67 4023.02 2340   7320   9990
#> 9  P.Undergrad       0      777 777    855.3  1522.43    1     95    353
#> 10 perc.alumni       0      777 777    22.74   12.39    0     13     21
#> 11    Personal       0      777 777  1340.64  677.07  250    850   1200
#> 12         PhD       0      777 777    72.66   16.33    8     62     75
#> 13  Room.Board       0      777 777  4357.53 1096.7  1780   3597   4200
#> 14   S.F.Ratio       0      777 777    14.09    3.96   2.5   11.5   13.6
#> 15    Terminal       0      777 777    79.7    14.72   24     71     82
#> 16   Top10perc       0      777 777    27.56   17.64    1     15     23
#> 17   Top25perc       0      777 777    55.8    19.8     9     41     54
#>        p75    max     hist
#> 1    2424   26330
#> 2    3624   48094
#> 3     600    2340
#> 4     902    6392
#> 5   10830   56233
#> 6    4005   31643
#> 7      78     118
#> 8   12925   21700
#> 9     967   21836
#> 10     31      64
#> 11   1700    6800
#> 12     85     103
#> 13   5050    8124
#> 14    16.5    39.8
#> 15     92     100
#> 16     35      96
#> 17     69     100
```

ii.

```r
pairs(College[, 1:5])
```

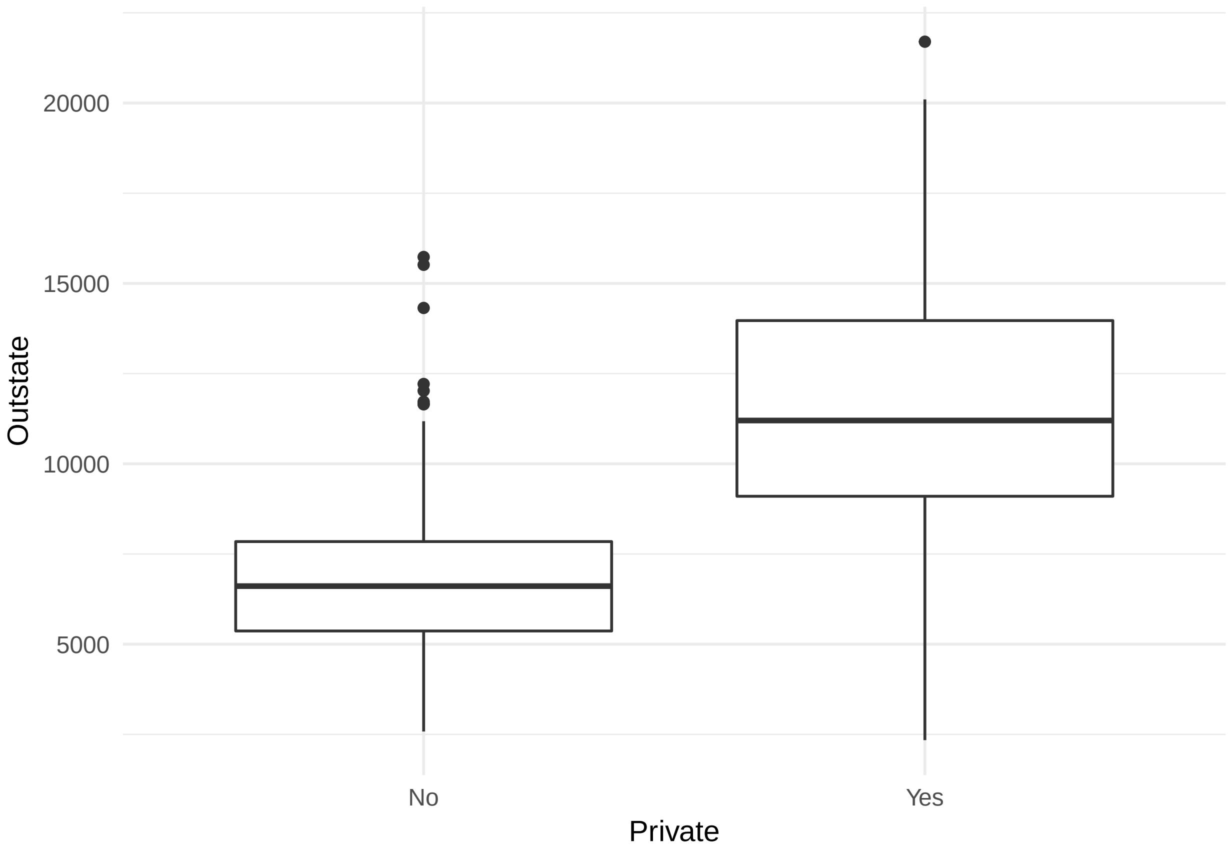



```r
ggpairs(College[, 1:5],
        aes(color = Private)) %>%
  print(progress = FALSE)
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



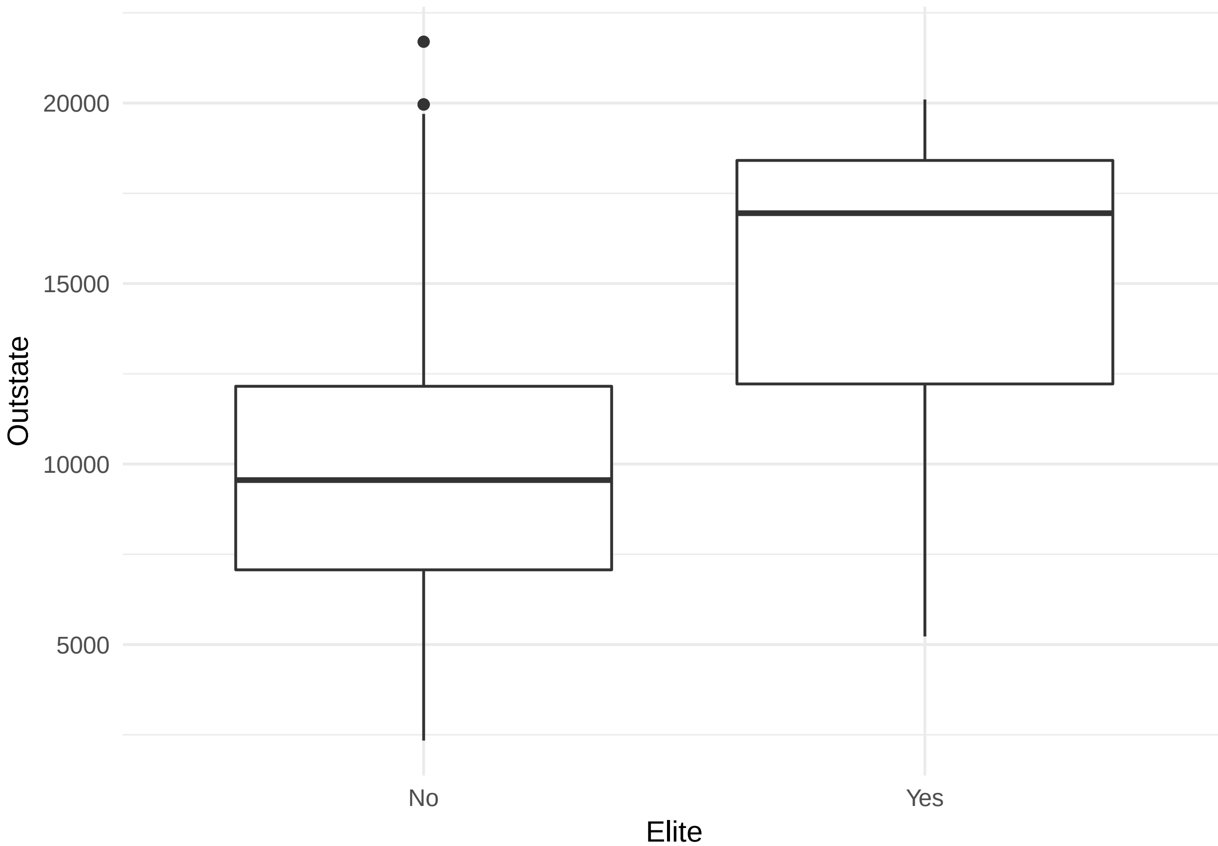
iii.

```r
ggplot(College, aes(x = Private, y = Outstate)) +
  geom_boxplot()
```



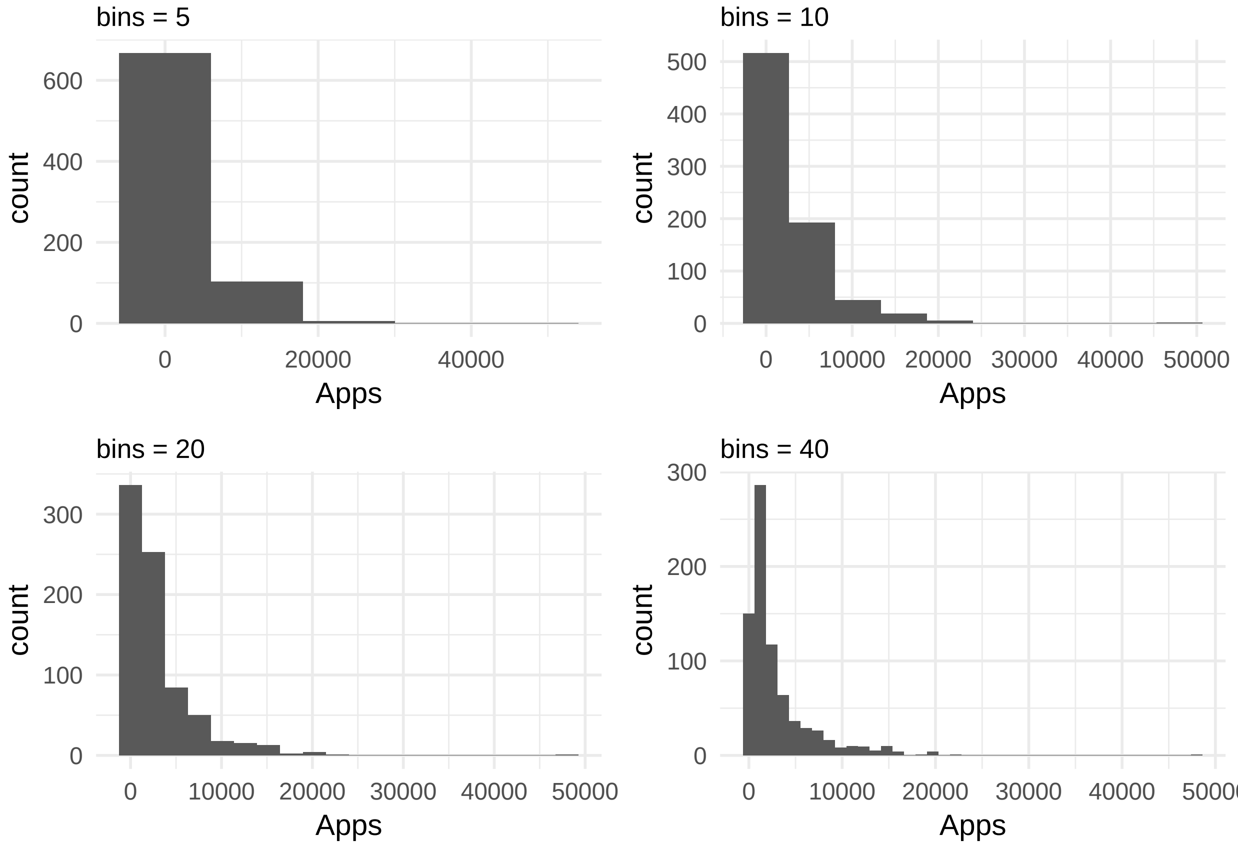
iv.

```
College %<>%
  mutate(Elite = if_else(Top10perc > 50,
                         "Yes", "No") %>%
           factor())

College %>%
  count(Elite)
#> # A tibble: 2 x 2
#>    Elite      n
#>   <fctr> <int>
#> 1 No       699
#> 2 Yes       78

ggplot(College, aes(x = Elite, y = Outstate)) +
  geom_boxplot()
```

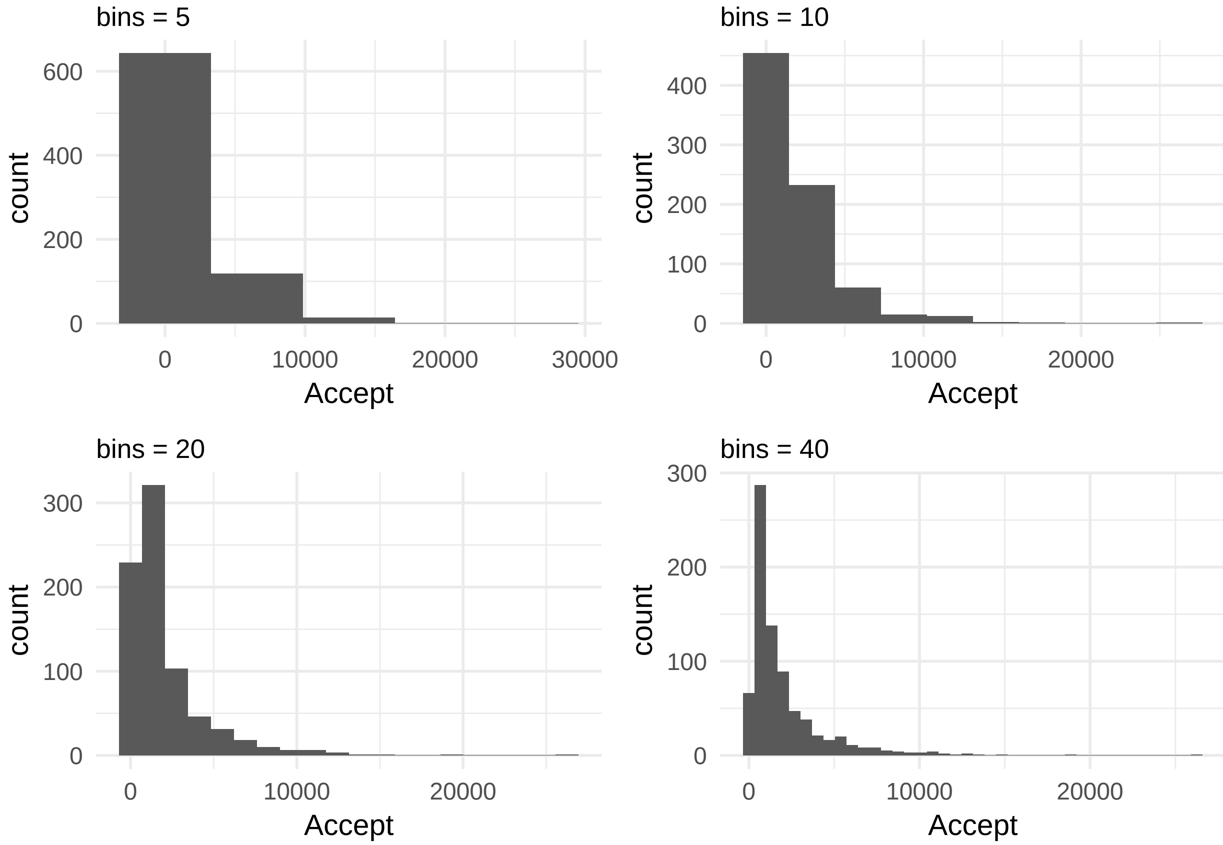


v.

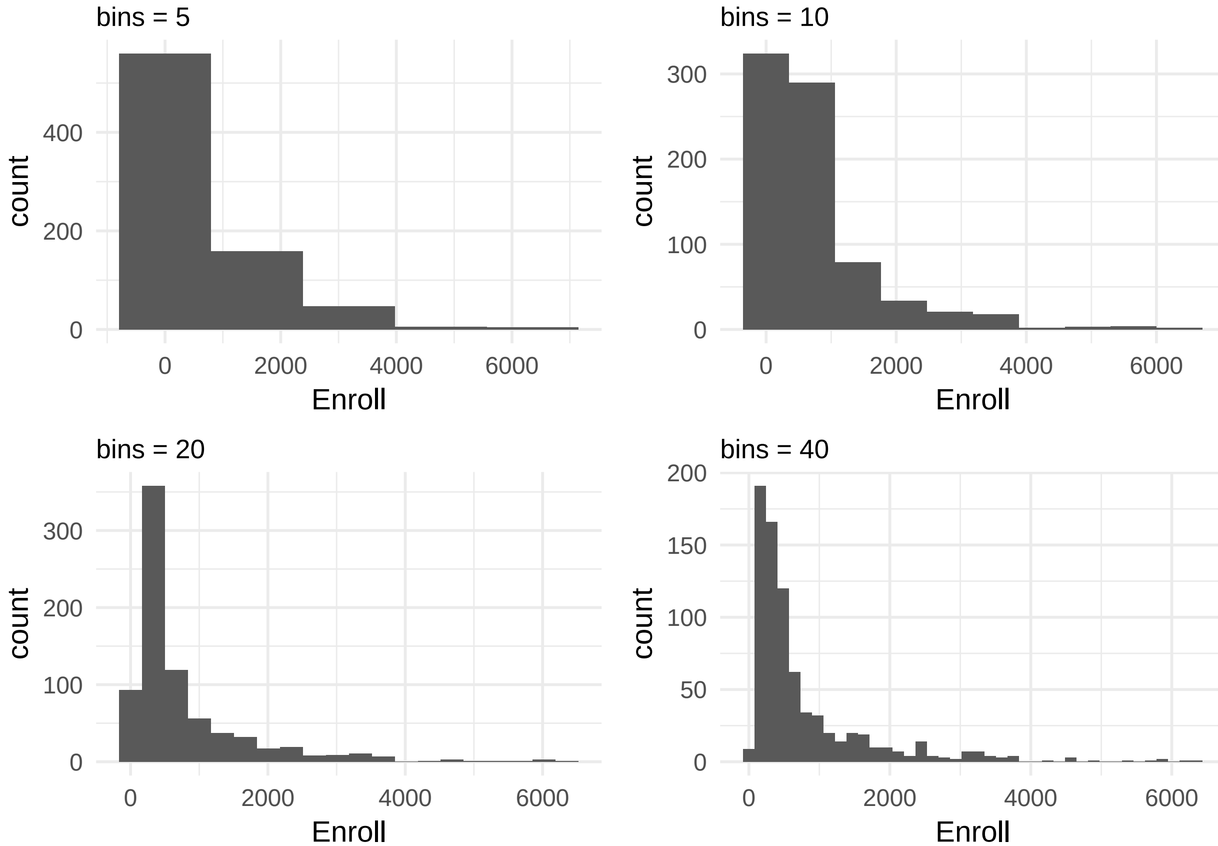
```
bins <- c(5, 10, 20, 40)
flex_hist <- function(var, bins, df) {
  var <- rlang::sym(var)
  map(bins,
      ~ ggplot(df, aes_((var))) +
        geom_histogram(bins = .x) +
        labs(subtitle = str_c("bins = ", .x)))
}

plots <- map(c("Apps", "Accept", "Enroll", "Grad.Rate"),
             flex_hist, bins = bins, df = College)

map(plots,
    ~ plot_grid(plotlist = .x, ncol = 2))
#> [[1]]
```


bins = 5
bins = 10
bins = 20
bins = 40
count
Apps
0
200
400
600
20000
40000
100
300
500
10000
30000
50000

```
#>
#> [[2]]
```

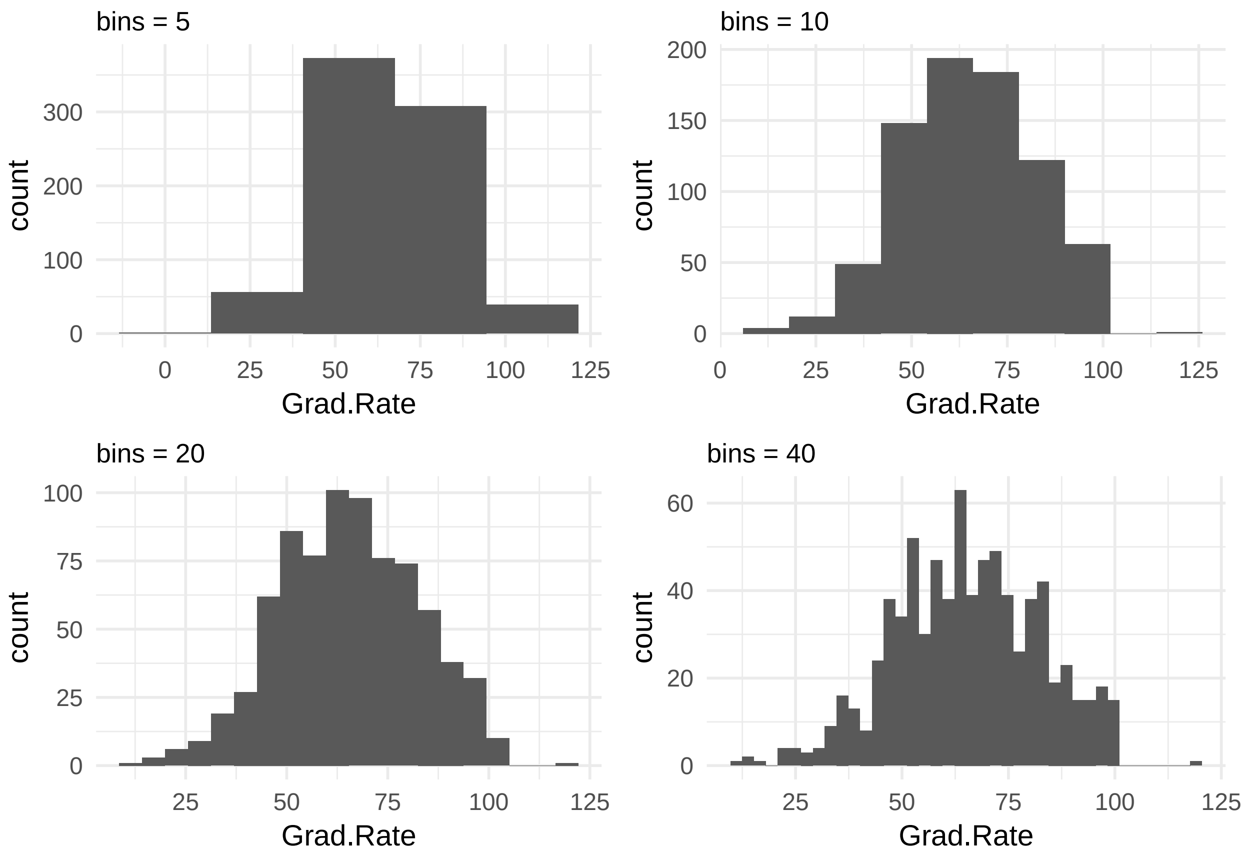

bins = 5
bins = 10
bins = 20
bins = 40
count
Accept
0
200
400
600
100
300
10000
20000
30000

```
#>
#> [[3]]
```



bins = 5
bins = 10
bins = 20
bins = 40
count
Enroll
0
200
400
100
300
50
150
2000
4000
6000


```
#>
#> [[4]]
```



vi. To be amended

(9)

```
Auto <- as_tibble(Auto)

Auto %>% filter_all(any_vars(is.na(.)))
#> # A tibble: 0 x 9
#> # ... with 9 variables: mpg <dbl>, cylinders <dbl>, displacement <dbl>,
#> #   horsepower <dbl>, weight <dbl>, acceleration <dbl>, year <dbl>,
#> #   origin <dbl>, name <fctr>
```

No `NA`s in the data.

a.

```
glimpse(Auto)
#> Observations: 392
#> Variables: 9
#> $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 1...
#> $ cylinders    <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6...
#> $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390,...
#> $ horsepower   <dbl> 130, 165, 150, 150, 140, 198, 220, 215, 225, 190,...
#> $ weight       <dbl> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4...
#> $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10....
#> $ year         <dbl> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 7...
#> $ origin       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1...
#> $ name         <fctr> chevrolet chevelle malibu, buick skylark 320, pl...
```

`origin` is the only qualitative predictor. `name` is rather a row id, all other variables are quantitative.

```
Auto %<>%
  mutate(origin = factor(origin,
                         levels = 1:3,
                         labels = c("American", "European", "Japanese")))
```

b. see (c)

c.

```
lower.boundary <- function(x) {range(x, na.rm = TRUE)[1]}
upper.boundary <- function(x) {range(x, na.rm = TRUE)[2]}

Auto %>%
  summarise_if(is.numeric,
               funs("mean", "sd", "lower.boundary", "upper.boundary")) %>%
  gather() %>%
  separate(key, into = c("variable", "measure"), sep = "_") %>%
  spread(measure, value) %>%
  dplyr::select(variable, ends_with("boundary"), everything())
#> # A tibble: 7 x 5
#>   variable     lower.boundary upper.boundary    mean     sd
#> * <chr>                 <dbl>          <dbl>   <dbl>  <dbl>
#> 1 acceleration           8.00           24.8    15.5   2.76
#> 2 cylinders              3.00            8.00    5.47  1.71
#> 3 displacement          68.0           455     194   105
#> 4 horsepower            46.0           230     104    38.5
#> 5 mpg                    9.00           46.6    23.4   7.81
#> 6 weight              1613            5140    2978   849
#> 7 year                  70.0            82.0    76.0   3.68
```

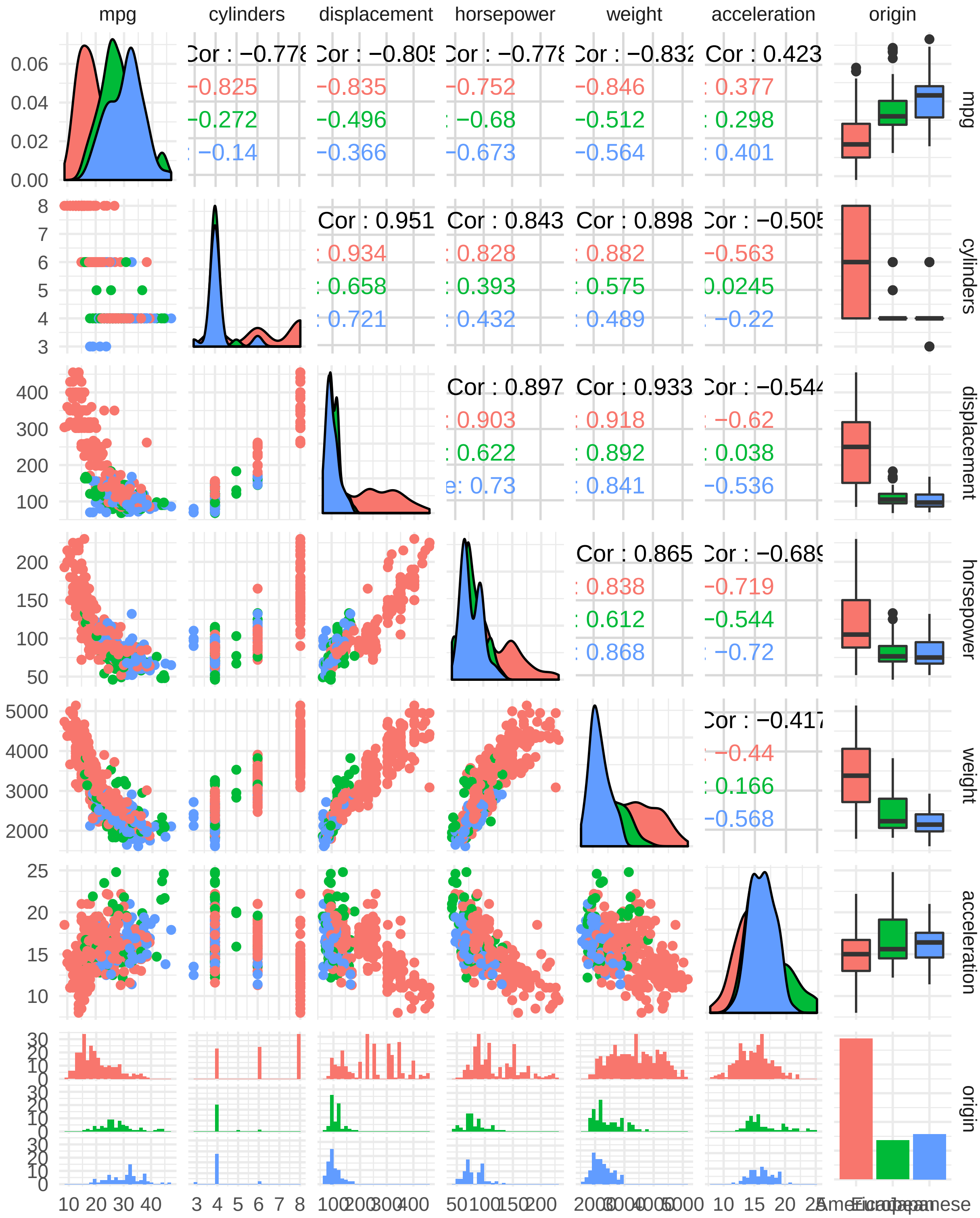
d.

```
Auto %>%
  slice(-(10:85)) %>%
  summarise_if(is.numeric,
               funs("mean", "sd", "lower.boundary", "upper.boundary")) %>%
  gather() %>%
  separate(key, into = c("variable", "measure"), sep = "_") %>%
  spread(measure, value) %>%
  dplyr::select(variable, ends_with("boundary"), everything())
#> # A tibble: 7 x 5
#>   variable     lower.boundary upper.boundary    mean     sd
#> * <chr>                 <dbl>          <dbl>   <dbl>  <dbl>
#> 1 acceleration           8.50           24.8    15.7   2.69
#> 2 cylinders              3.00            8.00    5.37  1.65
#> 3 displacement          68.0           455     187    99.7
#> 4 horsepower            46.0           230     101    35.7
#> 5 mpg                   11.0            46.6    24.4   7.87
#> 6 weight              1649            4997    2936   811
#> 7 year                  70.0            82.0    77.1   3.11
```

e.

```
Auto %>%
  dplyr::select(-year, -name) %>%
```

```r
  ggpairs(aes(color = origin)) %>%
  print(progress = FALSE)
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



f. `cylinders`, `displacement`, `horsepower` and `weight` show a strong negative correlation with `mpg` so these should be included in the model.

(10)

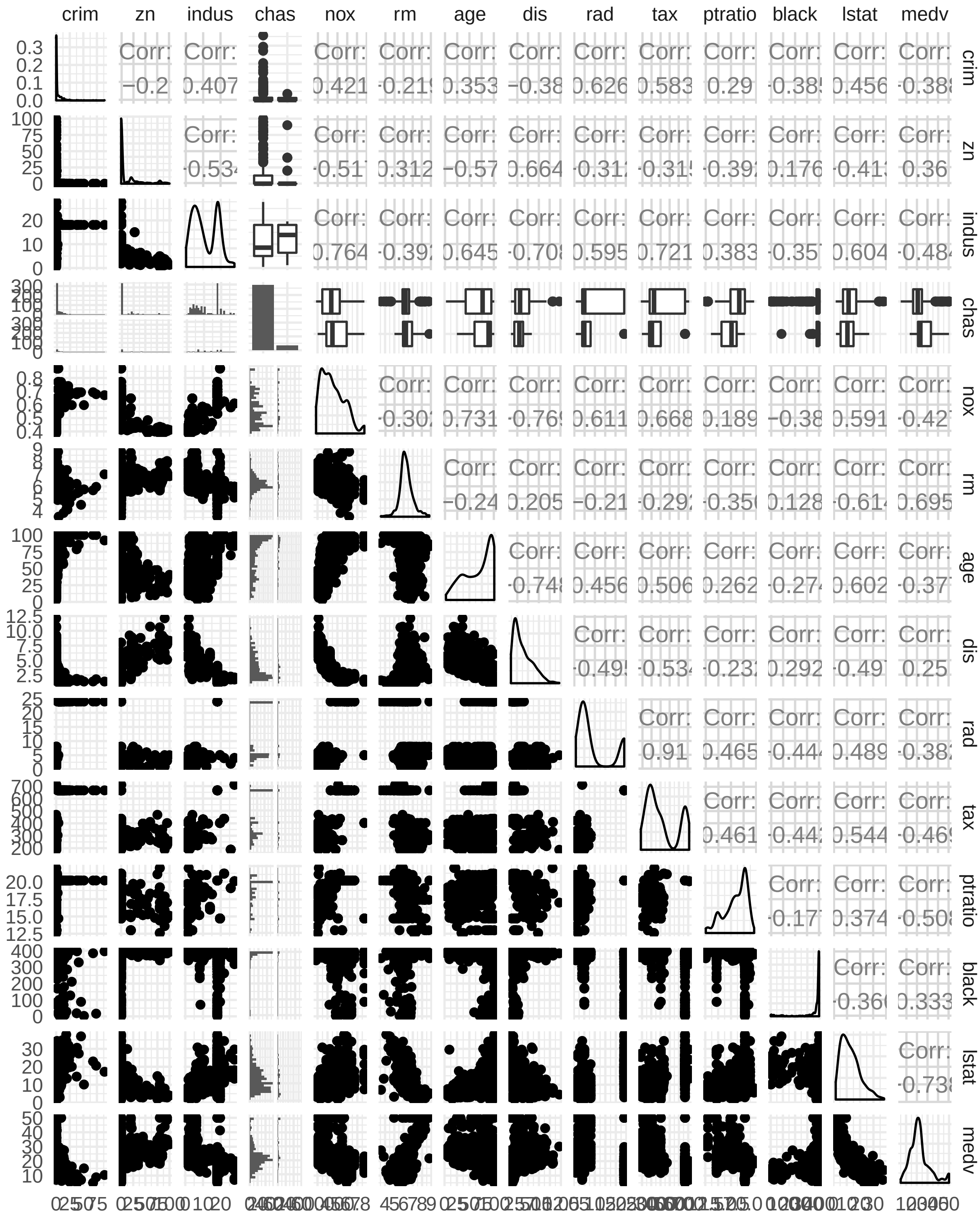
a.

```r
Boston <- as_tibble(Boston) %>%
  mutate(chas = factor(chas, levels = 0:1, labels = c("otherwise", "river bound")))
dim(Boston)
#> [1] 506  14
```

Rows represent suburbs, columns represent characteristics of these suburbs.

b.

```r
Boston %>%
  ggpairs() %>%
  print(progress = FALSE)
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

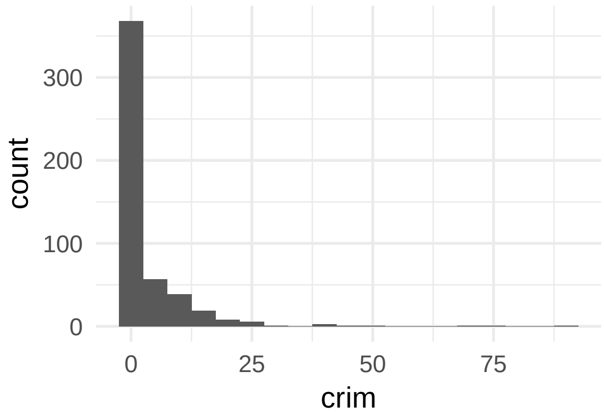


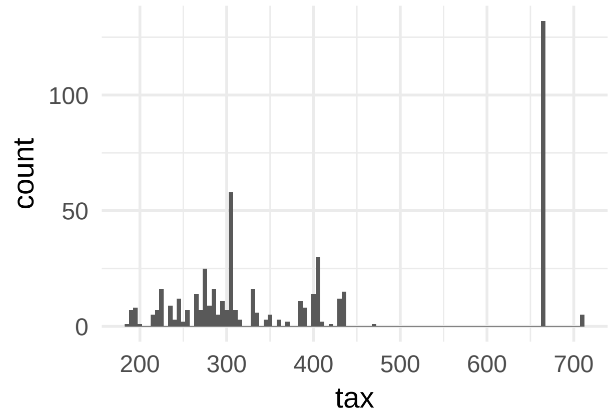
c. Basically all predictors are associated with `crim`, but not in a linear way. Hence, the correlations are rather low.

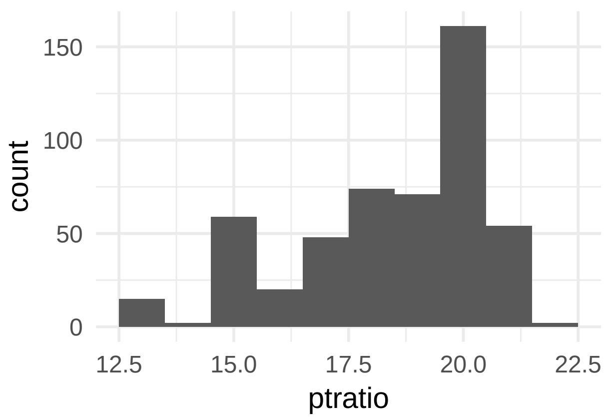
d.

```
Boston %>%
  top_n(n = 10, wt = crim)
```

```
#> # A tibble: 10 x 14
#>      crim    zn indus chas         nox    rm   age   dis   rad   tax ptratio
#>     <dbl> <dbl> <dbl> <fctr>     <dbl> <dbl> <dbl> <dbl> <int> <dbl>   <dbl>
#>  1  89.0     0  18.1 otherwise 0.671  6.97  91.9  1.42    24   666    20.2
#>  2  38.4     0  18.1 otherwise 0.693  5.45 100    1.49    24   666    20.2
#>  3  41.5     0  18.1 otherwise 0.693  5.53  85.4  1.61    24   666    20.2
#>  4  67.9     0  18.1 otherwise 0.693  5.68 100    1.43    24   666    20.2
#>  5  51.1     0  18.1 otherwise 0.597  5.76 100    1.41    24   666    20.2
#>  6  28.7     0  18.1 otherwise 0.597  5.16 100    1.59    24   666    20.2
#>  7  45.7     0  18.1 otherwise 0.693  4.52 100    1.66    24   666    20.2
#>  8  25.9     0  18.1 otherwise 0.679  5.30  89.1  1.65    24   666    20.2
#>  9  73.5     0  18.1 otherwise 0.679  5.96 100    1.80    24   666    20.2
#> 10  37.7     0  18.1 otherwise 0.679  6.20  78.7  1.86    24   666    20.2
#> # ... with 3 more variables: black <dbl>, lstat <dbl>, medv <dbl>

map2(rlang::quos(crim, tax, ptratio), c(5, 5, 1),
     ~ ggplot(Boston, aes_(.x)) +
       geom_histogram(binwidth = .y)) %>%
  plot_grid(plotlist = .)
```







e.

```
Boston %>%
  count(chas)
#> # A tibble: 2 x 2
#>   chas             n
#>   <fctr>       <int>
```

```
#> 1 otherwise     471
#> 2 river bound    35
```

f.

```
Boston %>%
  summarise(median(ptratio))
#> # A tibble: 1 x 1
#>   `median(ptratio)`
#>               <dbl>
#> 1              19.0
```

g.

```
Boston %>%
  mutate(low_medv = if_else(medv == min(medv), TRUE, FALSE)) %>%
  group_by(low_medv) %>%
  summarise_if(is.numeric, mean)
#> # A tibble: 2 x 14
#>   low_m~  crim    zn indus   nox    rm   age   dis   rad   tax ptra~ black
#>   <lgl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 F       3.42  11.4  11.1 0.554  6.29  68.5  3.80  9.49   407  18.4   357
#> 2 T      53.1    0    18.1 0.693  5.57 100    1.46 24.0    666  20.2   391
#> # ... with 2 more variables: lstat <dbl>, medv <dbl>
```

h.

```
map_df(list("rm > 7" = 7, "rm > 8" = 8),
       ~ Boston %>%
         filter(rm > .x) %>%
         nrow())
#> # A tibble: 1 x 2
#>   `rm > 7` `rm > 8`
#>      <int>    <int>
#> 1       64       13

Boston %>%
  group_by(rm > 8) %>%
  summarise_if(is.numeric, mean)
#> # A tibble: 2 x 14
#>   `rm >~  crim    zn indus   nox    rm   age   dis   rad   tax ptra~ black
#>   <lgl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 F      3.69   11.3 11.2  0.555  6.23  68.5  3.80  9.60   410  18.5   356
#> 2 T      0.719  13.6  7.08 0.539  8.35  71.5  3.43  7.46   325  16.4   385
#> # ... with 2 more variables: lstat <dbl>, medv <dbl>
```