Module 2: Recommended Exercises

alexaoh

17.01.2021

Problem 1

a)

Describe a real-life application in which classification might be useful. Identify the response and the predictors. Is the goal inference or prediction?

A real-life application of classification might be, for example, in deciding whether a special type of diet might lead to heart disease. In this case, the predictors would be the different types of diets and the response would be whether or not each person in the study develops cardiovascular disease or not. The goal might be both prediction and inference: One might want to predict whether a person has high risk of heart disease based on his/her diet, or one might want to learn which types of diets are most dangerous in the cardiovascular sense.

b)

Describe a real-life application in which regression might be useful. Identify the response and the predictors. Is the goal inference or prediction?

A real-life application of regression might be, for example, to predict how the standings in a football league will be at the end of the season. In this case, the predictors would be standings in the previous seasons, historical trends, stats of newly transferred players in each team, earlier results between each teams or others. The response would be the placements of each team in the league. The goal of this application is prediction.

Problem 2

Take a look at Figure 2.9 in the course book (p.31).

a)

Will a flexible or rigid method typically have the highest test error?

Somewhere in between a very flexible and a very rigid model will often be the sweet spot. Both a flexible and a rigid model will typically have high test error. Which of these is highest depends on the distribution of the data. However, between the models chosen in the figure, it is apparent that the more rigid method (linear regression) has the highest test error, but this is specific to this example.

b)

Does a small variance imply that the data has been under- or overfit?

A large variance could imply that the data has been overfit, because more flexible statistical methods have higher variance. This is the result of the flexible method following the observations very closely, which leads to a high variance, since the estimated function $\hat{f}$ will change a lot if the observations change. This overfitting is observed with increasing flexibility in the figure, because the mean squared error increases for the test data, despite the decrease of the mean squared error for the training data. One can say that the flexible model tries too hard to find patterns in the data, and consequently picks up patterns that are not to be found in reality (these are caused by random chance and not by properties of the unknown function f).

c)

Relate the problem of over- and underfitting to the bias-variance trade-off.

The bias-variance trade-off says that, in order to minimize the expected test error, one needs to select statistical models that achieves low variance and low bias. A very flexible method will achieve low bias, but high variance, while a very rigid method will achieve low variance, but high bias. When the data is overfit, the variance becomes too large, despite the fact that the bias is small. In this case, the variance is "overpowering" the decrease in bias, which means that the expected test error increases. Similarly, when the data is underfit, the variance is low but the bias is large. In this case, the bias is too large compared to the low variance, and the expected test error increases. This is why a model which has the "right amount" of flexibility often is the best way to go when the goal is to minimize the expected test error.

Problem 3 – Exercise 2.4.9 from ISL textbook (modified)

This exercise involves the `Auto` dataset from the `ISLR` library. Load the data into your R session by running the following commands:

```r
library(ISLR)
data(Auto)
```

a)

View the data. What are the dimensions of the data? Which predictors are quantitative and which are qualitative?

```r
dim(Auto) # Dimensions.
```

```
#> [1] 392   9
```

```r
summary(Auto)
```

```
#>       mpg          cylinders      displacement     horsepower        weight    
#>  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613  
#>  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225  
#>  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804  
#>  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978  
#>  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615  
#>  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140  
#>                                                                                
#>   acceleration        year           origin                      name    
#>  Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5  
#>  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5  
#>  Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5  
#>  Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4  
#>  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4  
#>  Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4  
#>                                                  (Other)           :365
```

```r
sapply(Auto, class) # Makes it more obvious which predictors are qualitative and quantitative.
```

```
#>          mpg    cylinders displacement   horsepower       weight acceleration 
#>    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric" 
#>         year       origin         name 
#>    "numeric"    "numeric"     "factor"
```

All predictors are quantitative, except for 'name', which is qualitative.

b)

What is the range (min, max) of each quantitative predictor? Hint: use the `range()` *function. For more advanced users, check out* `sapply()`*.*

```
# Remember to import dplyr in setup for the first variant to work.
# Two different methods of removing the categorical variable.
quant <- Auto %>% select(-name) # Using dplyr.
quant2 <- Auto[, -c(9)] # Using regular R.
identical(quant, quant2) # Sidenote: Shows that the two methods give the same result.
```

```
#> [1] TRUE
```

```
sapply(quant, range)
```

```
#>       mpg cylinders displacement horsepower weight acceleration year origin
#> [1,]  9.0         3           68         46   1613          8.0   70      1
#> [2,] 46.6         8          455        230   5140         24.8   82      3
```

c)

What is the mean and standard deviation of each quantitative predictor?

```
sapply(quant, mean) # Mean.
```

```
#>          mpg    cylinders displacement   horsepower       weight acceleration
#>    23.445918     5.471939   194.411990   104.469388  2977.584184    15.541327
#>         year       origin
#>    75.979592     1.576531
```

```
sapply(quant, sd) # Standard deviation.
```

```
#>          mpg    cylinders displacement   horsepower       weight acceleration
#>    7.8050075    1.7057832  104.6440039   38.4911599  849.4025600    2.7588641
#>         year       origin
#>    3.6837365    0.8055182
```

d)

Now, make a new dataset called `ReducedAuto` *where you remove the 10th through 85th observations. What is the range, mean and standard deviation of the quantitative predictors in this reduced set?*

```
ReducedAuto <- Auto[-c(10:85), ]
dim(ReducedAuto) # The rows have been removed.
```

```
#> [1] 316   9
```

```
quant.ReducedAuto <- ReducedAuto %>% select(-name)
sapply(quant.ReducedAuto, range) # Range.
```

```
#>       mpg cylinders displacement horsepower weight acceleration year origin
#> [1,] 11.0         3           68         46   1649          8.5   70      1
#> [2,] 46.6         8          455        230   4997         24.8   82      3
```

```
sapply(quant.ReducedAuto, mean) # Mean.
```

```
#>          mpg    cylinders displacement   horsepower       weight acceleration
#>    24.404430     5.373418   187.240506   100.721519  2935.971519    15.726899
```

```
#>         year       origin
#>    77.145570     1.601266
```

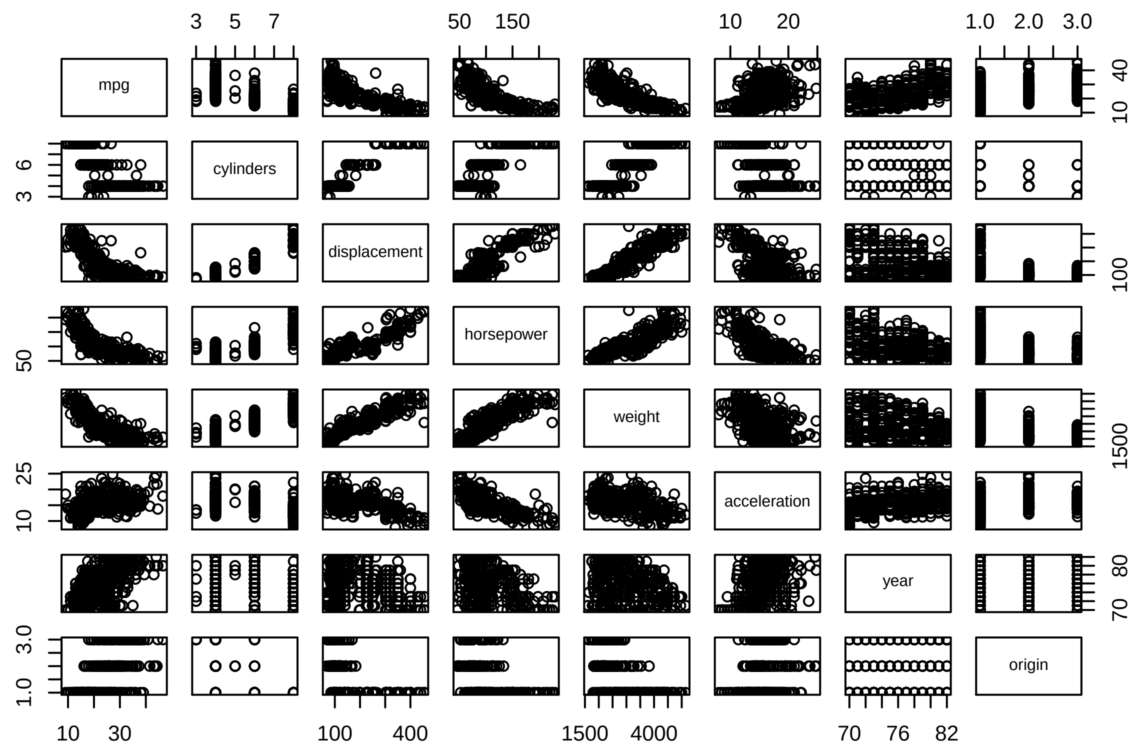
```r
sapply(quant.ReducedAuto, sd) # Standard deviation.
```

```
#>          mpg    cylinders displacement   horsepower       weight acceleration
#>     7.867283     1.654179    99.678367    35.708853   811.300208     2.693721
#>         year       origin
#>     3.106217     0.819910
```

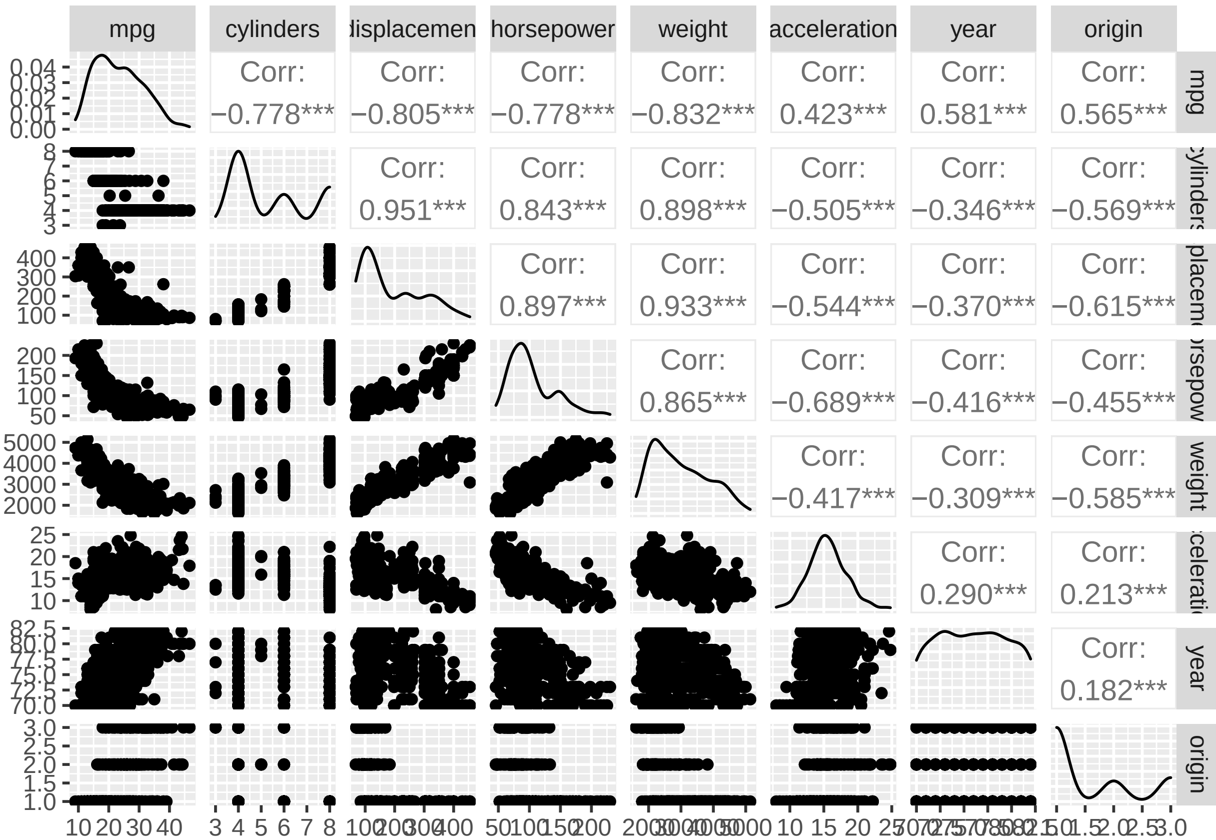
e)

Using the full dataset, investigate the quantitative predictors graphically using a scatterplot. Do you see any strong relationships between the predictors? Hint: try out the `ggpairs()` *function from the* `GGally` *package.*

```r
library(GGally)
pairs(quant) # Regular pairs plot.
```




```r
ggpairs(quant)
```



Based on the scatter plots, some relationships between the quantitative predictors seem to be stronger than others. It looks like the following pairs of predictors are highly correlated

- Displacement and weight
- Horsepower and weight
- Displacement and horsepower

f)

Suppose we wish to predict gas milage (`mpg`) on the basis of the other variables (both quantitative and qualitative). Make some plots showing the relationships between `mpg` and the qualitative predictors (hint: `geom_boxplot()`). Which predictors would you consider helpful when predicting `mpg`?

g)

The correlation of two variables X and Y are defined as

$$\mathrm{cor}(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y}.$$

Both the correlation matrix and covariance matrix are easily assessed in R with the `cor()` and `cov()` functions. Use only the covariance matrix to find the correlation between `mpg` and `displacement`, `mpg` and `horsepower`, and `mpg` and `weight`. Do your results coincide with the correlation matrix you find using `cor(Auto[,quant])`?

```
quant = c(1,3,4,5,6,7)
covMat = cov(Auto[,quant])
```

Problem 4 – Multivariate normal distribution

The pdf of a multivariate normal distribution is on the form

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\},$$

where $\mathbf{x}$ is a random vector of size $p \times 1$, $\boldsymbol{\mu}$ is the mean vector of size $p \times 1$ and $\boldsymbol{\Sigma}$ is the covariance matrix of size $p \times p$.

a)

Use the `mvrnorm()` function from the `MASS` library to simulate 1000 values from multivariate normal distributions with

i)

$$\boldsymbol{\mu} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

```r
library(MASS)
```

```
#>
#> Attaching package: 'MASS'
#> The following object is masked from 'package:dplyr':
#>
#>     select
```

```r
samples.1 <- mvrnorm(n = 1000, mu = c(2, 3), Sigma = matrix(c(1, 0, 0, 1), nrow = 2))
```

ii)

$$\boldsymbol{\mu} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix},$$

```r
samples.2 <- mvrnorm(n = 1000, mu = c(2, 3), Sigma = matrix(c(1, 0, 0, 5), nrow = 2))
```

iii)

$$\boldsymbol{\mu} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix},$$

```r
samples.3 <- mvrnorm(n = 1000, mu = c(2, 3), Sigma = matrix(c(1, 2, 2, 5), nrow = 2))
```

iv)

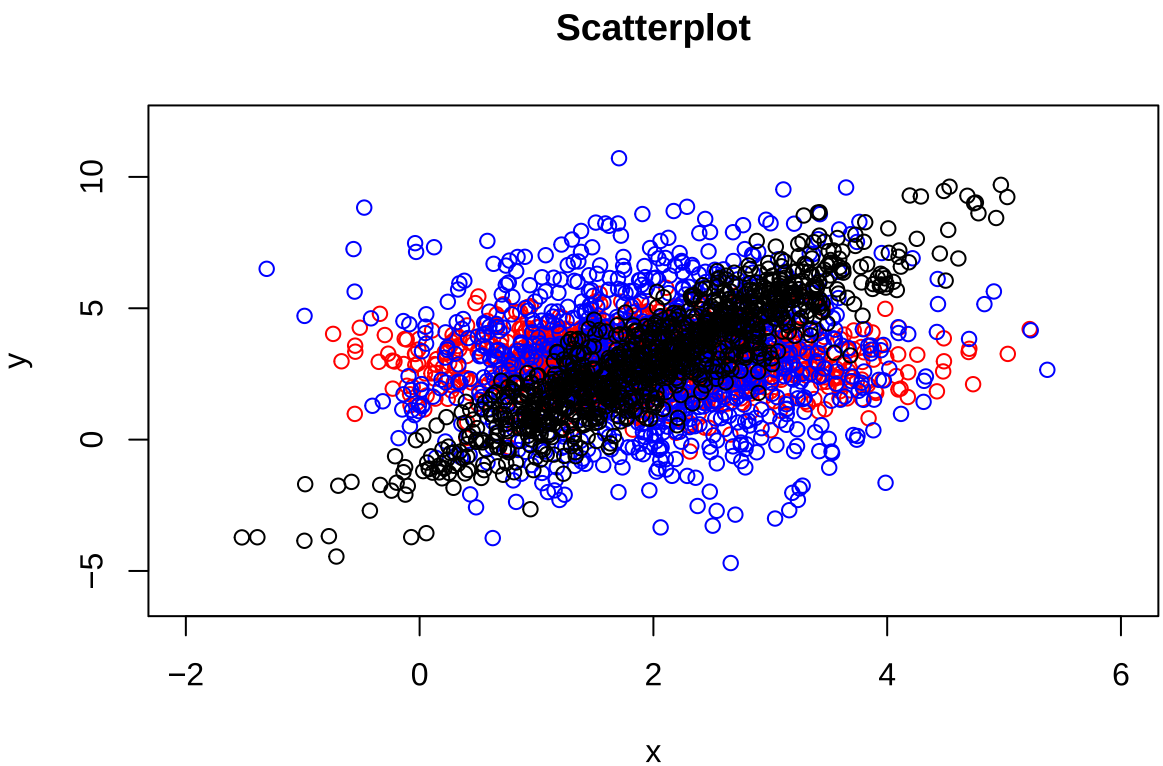
$$\boldsymbol{\mu} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix}.$$

```r
#samples.4 <- mvrnorm(n = 1000, mu = c(2, 3), Sigma = matrix(c(1, -2, -2, 1), nrow = 2))
# This does not work, since Sigma is not p.d.
```

b)

Make a scatterplot of the four sets of simulated datasets. Can you see which plot belongs to which distribution?

```r
plot(NULL, NULL, main = "Scatterplot", xlim = c(-2, 6), ylim = c(-6, 12), xlab = "x", ylab = "y")
points(samples.1, col = "red")
points(samples.2, col = "blue")
points(samples.3)
```



It is apparent that the black dots belong to the distribution in iii), since these points are clearly correlated, which is not the case for the two first distributions. Furthermore, the blue points correspond to distribution ii), since the variance is larger in y. The red points then correspond to i), which is a "standard Gaussian" around (2, 3).

Problem 5 – Theory and practice: training and test MSE; bias-variance