Problem Set 14

David Rügamer, Julia Terhart, Philipp Kopper

20 July 2020

Resources

1) Read the chapters on loops, conditionals, and functions in R4DS. The chapter can be found on Moodle.
2) Optionally, complete the DataCamp Units provided on the Moodle page for this topic.
3) Accept the invitation to the assignment of this problem set: https://classroom.github.com/a/QYxPXzrX

Application

Have a look at the following data set:

```
library(MASS)
head(round(Boston, 3))
```

```
##    crim zn indus chas   nox    rm  age   dis rad tax ptratio  black lstat medv
## 1 0.006 18  2.31    0 0.538 6.575 65.2 4.090   1 296    15.3 396.90  4.98 24.0
## 2 0.027  0  7.07    0 0.469 6.421 78.9 4.967   2 242    17.8 396.90  9.14 21.6
## 3 0.027  0  7.07    0 0.469 7.185 61.1 4.967   2 242    17.8 392.83  4.03 34.7
## 4 0.032  0  2.18    0 0.458 6.998 45.8 6.062   3 222    18.7 394.63  2.94 33.4
## 5 0.069  0  2.18    0 0.458 7.147 54.2 6.062   3 222    18.7 396.90  5.33 36.2
## 6 0.030  0  2.18    0 0.458 6.430 58.7 6.062   3 222    18.7 394.12  5.21 28.7
```

The Boston housing data set is one of the most frequently used data sets in machine learning and statistics. It contains information collected by the U.S Census Service concerning housing in the area of Boston. It was obtained from the StatLib archive (http://lib.stat.cmu.edu/datasets/boston) and has been used extensively throughout the literature to benchmark algorithms. The data was originally published by Harrison and Rubinfeld (for citation see `?Boston`). For details on the data call `?Boston`.

a) The data features categorical features which are, however, stored as integers. Use the `dplyr` package to change all categorical features to a `factor` with the levels provided in `?Boston`. Store the data set as `Boston_new`.

b) Assume you want to use the `Boston` data set to test a newly developed algorithm. Your algorithm is sensitive to the absolute values of features. Hence, it would always prefer these features which are large by construction. For example, a difference of 4 years in `age` is probably much less substantial than a difference of 0.4 in `nox`. To eliminate this sensitivity, one could univariately scale the numeric data. Univariate scaling simply subtracts the mean of a value and divides the value by its standard deviation. Write a for-loop that iterates over the columns of `Boston` and checks if the column is numeric. For the numeric columns apply scaling. For the categorical ones, leave them as they are. `Boston_new` should be scaled for the consequent questions.

Example: A vector reporting height in cm

```
## [1] 156 189 177 160 182 178
```

becomes (when scaling):

```
## [1] -1.3684541  1.1877149  0.2581989 -1.0586154  0.6454972  0.3356586
```

c) Use the `apply` family (`lapply`) family to (at least partly) vectorise your solution from b).

Transfer

Using the `Boston` data we want to find a linear model that explains the criminality in Boston. A social science research group already made some suggestions on possible models. However, they are not sure how to evaluate which model is actually the best. You as a Statistician suggest the use of adjusted R squared which is stored in the summary of a linear model as a model quality measure. The research group gives you a list containing ten different models. The list is stored as an RDS object in this repo. Load it into `R`.

a) Write a for-loop that iterates over all possible models and selects the best model – based on the adjusted R squared.

b) Create a function using your previous code which takes a list of linear models as input and returns the model with the highest adjusted R squared. Document your function using comments: Describe what the function does, what the inputs are, and what the output is. (See below.)

```
### FUNCTION
### THIS FUNCTION DOES SOMETHING
### Input:
### x1: a numeric vector which indicates the ....
### x2: a data.frame with X columns..
### Output:
### A list of .... representing...
myfunction <- function(x1, x2) {
  #TODO
}
```

b) Give your function an additional argument. The argument steers which criterion is chosen for model selection. Instead of adjusted R squared, you want to allow the use of R squared, too. Conditional on the argument, you select the respective model quality criterion. Name the additional argument `type`. Apply your new function to both possible inputs for `type` and show this way that the function works.

Hints: Make use of a conditional (`if`) construction. It is easiest if you base the condition on a string match. (i.e. `type` should be a character.)

The same research group is interested in finding similar suburbs in Boston. Even though you have some doubts if the presented search algorithm is really a good way to sample these suburbs, you agree to implement it according to their wishes: You define a suburb that you want to find similar suburbs for. For this suburb, you simply go through the data set and add observations to a `data.frame` if they are sufficiently similar to the observations already in the `data.frame`. As soon as you found 10 similar observations, you stop this search. Whether or not an observation is similar enough is steered by a threshold. Observations with a similarity larger than the threshold are accepted, others rejected.

c) We are going to determine similarity (or rather dissimilarity!) via the Manhattan distance (you may remember it from the beginning of the semester). The Manhattan distance does not work well with categorical data. Hence, use the `dplyr` package to drop the categorical data from the scaled data set.

d) Write a function that determines the dissimilarity of one single observation with a `data.frame`. Compute the Manhattan dissimilarity of the single observation to every other observation in the `data.frame` and compute the mean of it. Test your function with `observation = Boston_new[1,]` and `dataframe`

`= Boston_new`. Your function should take two inputs (`observation` and `dataframe`) and return a numeric value (the distance). Also, document your function. Name it `compute_distance`.

Hints: Make use of a for-loop *or* matrices. Also, make use of `rowSums` or `apply`.

e) Use a do-while-loop (or the `repeat` function) to find ten similar suburbs to `Boston_new[4, ]`. Simply go through the rows of the data frame from 1 to `nrow(Boston_new)`. Stop as soon as you found 10 similar suburbs. An observation can be seen as similar if its Manhattan distance to the data frame is less than 4.

Hints: Use the `rbind()` function. Make sure that you don't compare `Boston_new[4, ]` with the data.frame. Make sure that you stop if you went through the whole data set (even if you did not succeed in finding 10 suburbs). You will most likely need to use multiple nested `if`s. You will need to make use of indexing.

```r
sessionInfo()
```

```
## R version 4.0.0 (2020-04-24)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Catalina 10.15.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] MASS_7.3-51.6
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.0.0  magrittr_1.5    tools_4.0.0     htmltools_0.4.0
##  [5] yaml_2.2.1      Rcpp_1.0.5      stringi_1.4.6   rmarkdown_2.2
##  [9] knitr_1.28      stringr_1.4.0   xfun_0.14       digest_0.6.25
## [13] rlang_0.4.6     evaluate_0.14
```

You can hand in this problem set by the 31st of July to receive feedback.