Problem Set 11

David Rügamer, Julia Terhart, Philipp Kopper

29 June 2020

Resources

1) Accept the invitation to the assignment of this problem set: LINK

Application

We simulate some data now. We want to work with stock market data – we simulate three different stock prices for each single day in 2019. All follow a normal distribution and are somewhat correlated.

```
set.seed(06072020)
stock_data <- data.frame(
  time = as.Date("2019-01-01") + 0:364,
  x = rnorm(365, 15, 2)
)
stock_data$y = rnorm(365, 0.25 * stock_data$x + 15, 3)
stock_data$z = rnorm(365, 0.3 * stock_data$x + 20, 4)
```

a) Create a new `data.frame` which collapses the data so that it looks like the following (or similar). (Of course all the information of `stock_data` should also be in the new `data.frame`). Make use of the `tidyr` package.

```
data.frame(
  time = rep("2019-01-01", 3),
  stock = c("x", "y", "z"),
  price = c(stock_data$x[1:3])
)
```

```
##         time stock    price
## 1 2019-01-01     x 17.91941
## 2 2019-01-01     y 12.80538
## 3 2019-01-01     z 11.43205
```

b) Try to produce the same data set using the `reshape2` package.

c) According to Hadley Wickham (the author of R4DS) why is your new data set tidy?

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

d) Find the function from the `tidyr` package that reverts the previously constructed tidy data to `stock_data`.

e) Use the `dplyr` package to group the data by week and compute their mean. Use `ggplot2` to plot the weekly stock prices **nicely**.

Transfer

`OpenML` is an inclusive movement to build an open, organized, online ecosystem for machine learning. They facilitate a great environment for open machine learning. For machine learning, one typically needs data. For now, we use `OpenML` as a data warehouse. At later points in your study, you might come back here for a different purpose.

a) Get the data set we want to work with from `OpenML`: https://www.openml.org/d/41021

Read the data description (in fact the data is very interesting). Download it as `.csv` and load it into `R`. Store the data in your repository and make sure that using your code we can replicate your results. (In fact this is very important: In the Take-home exam we also expect this from you and if this does not work you will score 0 on the problem.)

b) The data is messy. Some types are not correct (e.g. character instead of numeric, no factors). Correct the types. Hint: For some columns like `Year` you could use both, factors or numerics.

c) There is missing data in the data set. Use the `dplyr` package to fill all the missing values with the mean of the respective column. For which columns may this be problematic?

```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
##
## 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_1.0.4.6    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 6th of July to receive feedback.