iris_data_set_vm6

Han Oostdijk (www.hanoostdijk.nl)

29 December, 2015

Introduction

This example shows how to print two tables side by side. It makes use of the function *print_table_sbs* that generates the LaTeX commands that are necessary for this. If this is usable depends on the size of the tables and your eyesight.

Utility functions and constants

```
numlist = 10 # number of observations in iris data set that is listed

print_table_sbs <- function (files,label,cap,caps,scalebox=0.85) {
# print xtables saved in files side-by-side
# derived from Marcin Kosiński
# http://stackoverflow.com/questions/23926671/side-by-side-xtables-in-rmarkdown
cat('\\begin{table}[ht]\n')
cat('\\centering\n')
for (i in 1:length(files)) {
  tc  = caps[[i]]
  tci = paste0(label,letters[i])
  f   = files[[i]]
  c = '\\subfloat[%s]{\\label{table:%s}\\scalebox{%.3f}{\\input{./%s}}}\\quad\n'
  cat(sprintf(c,tc,tci,scalebox,f))
}
cat(sprintf('\\caption{%s}\n',cap))
cat(sprintf('\\label{table:%s}\n',label))
cat('\\end{table}')
}
```

Explanation

We will create two tables in R and use package **xtable** to write the LaTeX representation to a tex-file. We use the iris data set as an example and will sort it in ascending and descending order (by the variable *Sepal.Length*). Because that is twice (nearly) the same code we have created the function *create_xtable* to do this. In the function we select only the first/last 10 observations. However the only important line is *print(xtable(. . . floating=FALSE)* .
The function *print_table_sbs* will take the two files and print them side by side if that fits. Otherwise the second table will be positioned below the first one. Table 1a on page 2 we will show the 10 observations with the smallest and Table 1b the ones with the greatest *Sepal.Length*.

```
library(xtable)
create_xtable<- function (decreasing,numlist,filename) {
  data(iris)
  tab = iris[order(iris$Sepal.Length,decreasing = decreasing),]
  print(xtable(tab[1:numlist,],row.names=F),
    file=paste0(filename,'.tex'),floating=FALSE)
}

create_xtable(F,numlist,filename='asc')
create_xtable(T,numlist,filename='desc')
```

```
files = c('asc','desc') # filenames (without suffix tex)
label = 'r1' # label (sublabels have suffix 1,2, ...)
cap   = 'first/last observations of iris data set sorted by \\mytextit{Sepal.Length}'
```

```
caps  = c('first', 'last')
print_table_sbs(files,label,cap,caps,scalebox=0.5)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
14	4.30	3.00	1.10	0.10	setosa
9	4.40	2.90	1.40	0.20	setosa
39	4.40	3.00	1.30	0.20	setosa
43	4.40	3.20	1.30	0.20	setosa
42	4.50	2.30	1.30	0.30	setosa
4	4.60	3.10	1.50	0.20	setosa
7	4.60	3.40	1.40	0.30	setosa
23	4.60	3.60	1.00	0.20	setosa
48	4.60	3.20	1.40	0.20	setosa
3	4.70	3.20	1.30	0.20	setosa

(a) first

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
132	7.90	3.80	6.40	2.00	virginica
118	7.70	3.80	6.70	2.20	virginica
119	7.70	2.60	6.90	2.30	virginica
123	7.70	2.80	6.70	2.00	virginica
136	7.70	3.00	6.10	2.30	virginica
106	7.60	3.00	6.60	2.10	virginica
131	7.40	2.80	6.10	1.90	virginica
108	7.30	2.90	6.30	1.80	virginica
110	7.20	3.60	6.10	2.50	virginica
126	7.20	3.20	6.00	1.80	virginica

(b) last

Table 1: first/last observations of iris data set sorted by *Sepal.Length*

```
cat(paste(' #produced',ref_tab('r1',prefix='')))
```

#produced Table 1 on page 2