

Refactoring the **xtable** package

Daniel Geals

Supervised by: Paul Murrell and David Scott

November 18, 2014

Outline

What is **xtable**

Issues with **xtable**

Refactoring

Testing Procedures

Conclusion

xtable example: R code

```
library(xtable)

#make some random data
words = rep(c("low", "medium", "high") ,3)
x = data.frame(var1 = rnorm(n = 9, mean = 10, sd = 2),
               var2 = rnorm(n = 9, mean = 5, sd = 15),
               var3 = runif(n = 9, min = -4, max = 4),
               var4 = rnorm(n = 9, mean = 0, sd = 100),
               factor = factor(sample(words)))

#create LaTeX table with xtable
tbl = xtable(x, digits = 2,
             caption = "data frame of random data
                        with booktabs option")
align(tbl) = c("|lr|r|r|l|l")
print(tbl, booktabs = TRUE)
```

xtable example: LaTeX output

```
\begin{table}[ht]
\centering
\begin{tabular}{|lr|r|r|r|l|}
\toprule
& var1 & var2 & var3 & var4 & factor \\
\midrule
1 & 14.13 & -7.14 & -2.11 & 112.00 & low \\
2 & 10.78 & -10.76 & -2.85 & 97.42 & low \\
3 & 8.43 & -21.03 & 2.60 & 38.07 & medium \\
4 & 8.71 & -12.83 & -2.73 & 145.81 & low \\
5 & 8.46 & 39.89 & -3.13 & 36.43 & medium \\
8 & 10.15 & 7.12 & 3.53 & -213.87 & high \\
9 & 12.09 & -22.52 & 2.21 & 94.50 & medium \\
\bottomrule
\end{tabular}
\caption{data frame of random data with booktabs option}
\end{table}
```

xtable example

	var1	var2	var3	var4	factor
1	14.13	-7.14	-2.11	112.00	low
2	10.78	-10.76	-2.85	97.42	low
3	8.43	-21.03	2.60	38.07	medium
4	8.71	-12.83	-2.73	145.81	low
5	8.46	39.89	-3.13	36.43	medium
8	10.15	7.12	3.53	-213.87	high
9	12.09	-22.52	2.21	94.50	medium

Table : data frame of random data with booktabs option

About **xtable**

- ▶ **xtable** is an R extension package that provides functionality for outputting \LaTeX and HTML code that when processed creates a table
- ▶ we can create a table with few lines of code but the package has many optional arguments for customisation
- ▶ **xtable** is geared toward \LaTeX table production with many \LaTeX specific arguments but few for HTML
- ▶ **xtable** is useful in the literate programming and reproducible document arena
- ▶ **xtable** mitigates risks inherent in creating \LaTeX and HTML tables 'by hand'

Outline

What is **xtable**

Issues with **xtable**

Refactoring

Testing Procedures

Conclusion

Issues with existing implementation

- ▶ `print.xtable`
- ▶ mixture of \LaTeX environments and HTML
- ▶ large number of function arguments
- ▶ 650 line function
- ▶ too many different operations taking place in one function
- ▶ package is difficult to maintain
- ▶ hard to add additional functionality
- ▶ not following proper programming practice

Outline

What is `xtable`

Issues with `xtable`

Refactoring

Testing Procedures

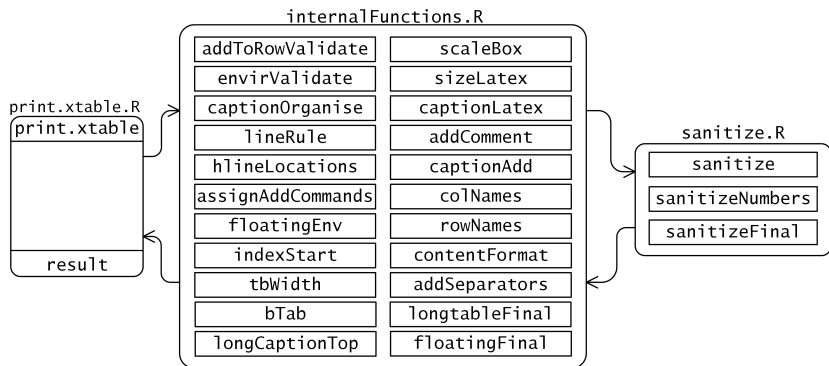
Conclusion

Refactoring definition

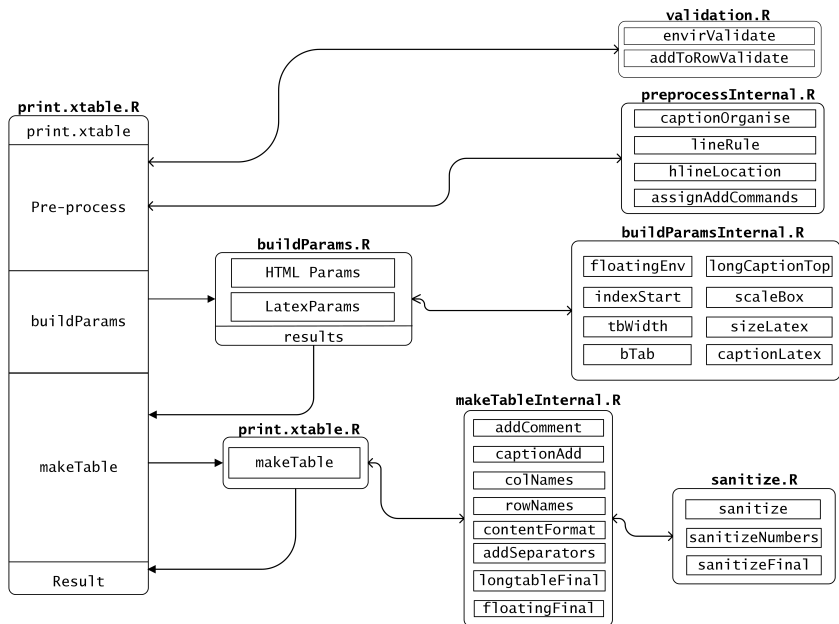
“Refactoring is the process of changing a software system in such a way that it does not alter the external behaviour of the code yet improves its internal structure. It is a disciplined way to clean up code that minimises chances of introducing bugs.”

— Martin Fowler, Refactoring

Creating new functions



Restructuring



Summary of changes made

- ▶ extract the lower level programmatic detail into internal functions
- ▶ give reasonably descriptive names to new functions
- ▶ minimise amount of activity in new functions
- ▶ remove some of the old comments, especially commented out code
- ▶ attempt to restructure the code with larger functions that call the newly created internal functions

Outline

What is `xtable`

Issues with `xtable`

Refactoring

Testing Procedures

Conclusion

Why we need testing

- ▶ package must take inputs and return outputs that are exactly the same as the existing version (keep external behaviour)
- ▶ large number of arguments that need to be tested as working
- ▶ combinations of these arguments also need to be tested as working
- ▶ invalid combinations of arguments need to be rejected with appropriate error messages
- ▶ having appropriate tests can form the basis of a development cycle

Testing procedures

Confirm changes have not broken the package

- ▶ use the provided gallery to test if modified functions create identical content
- ▶ use a diff utility to compare line by line between the old and new gallery

Confirm invalid input is returning appropriate errors

- ▶ create test suite using **testthat** package
- ▶ test that invalid inputs are being stopped by the validation functions

Ensure tests return informative output and are convenient to run

Testing example

```
test_that("Long table warning works", {  
  data(tli)  
  tli.table <- xtable(tli[1:10,])  
  #check floating and longtable combination gives warning  
  expect_that(print(tli.table, tabular.environment = "longtable",  
                    floating = TRUE), gives_warning())  
})
```

Outline

What is `xtable`

Issues with `xtable`

Refactoring

Testing Procedures

Conclusion

Refactored version

[illegible]

Conclusions

- ▶ function of issue has been refactored
- ▶ backwards compatibility
- ▶ future work

References

- Dahl, D. B. (2014). *xtable: Export tables to LaTeX or HTML*. R package version 1.7-3.
- Fowler, M., Beck, K., Brant, J., Opdyke, W., and Roberts, D. (2012). *Refactoring: Improving the Design of Existing Code*. Pearson Education.
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson Education.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Wickham, H. (2011). testthat: Get started with testing. *The R Journal*, 3:5–10.