**The {tabit} package**

This is a blog post announcing the brand new micro package {tabit} that just made it to CRAN.

Thanks to all CRAN people 

**Motivation**

{tabit} is a package that is about making simple tabulations simple.  
My motivation for writing this package was the realization that I was actually missing the way I could do tabulation in Stata: Easily getting an idea of the data very fast.

While R of cause has an tabulation function build in, I was always struggling with  
getting it to show what I wanted without specifying to many arguments.

The way I want it to work:

* I want frequencies.
* I want percentages.
* I do not want it to ignore NAs – never ever, no way.
* I want to see how thinks look like without NAs.
* I want it in a format that is easy to use later on or re-use for non interactive data glancing tasks.
* I want results to be sorted by decreasing frequency.
* I want it to be generic so I can make it work for vectors and lists and data.frames and all the things that might come up in the future.
* I want it to be configurable via parameters.
* I really do not want to have to touch those parameters ever.
* I want it to have zero dependencies.
* I want it all, I want it know

Over the last years I realized I was rewriting kind of the same function over and over again for projects and also for packages. After having gone through some iterations I am now quite happy with the outcome – though little code it might be.

**Giving it a try**

At the moment only one-dimensional tables are implemented multidimensional tabulations are planned but a well balanced design for a function still is under development.

Lets have a demo using the built in *“New York Air Quality”* data set. The data set consists of several variables most notably some containing missing values. The variable of interest Solar.R measures solar radiation in [Langleys](https://en.wikipedia.org/wiki/Langley_(unit)).

To get a quick overview I round the radiation measures to the nearest hundreds and use ti\_tab1() to get a frequency table. The result of a call to ti\_tab1() is a data.frame with on line per variable value, sorted by decreasing frequencies and including frequencies for missing values as well per default. Since percentages differ depending on whether or not missing values (NA) are included or not there is one column  
excluding NAs and one including them.

library(tabit)

ti\_tab1(

x = round(airquality$Solar.R, -2)

)

## value count pct pct\_all

## 3 200 50 34.25 32.68

## 4 300 45 30.82 29.41

## 2 100 34 23.29 22.22

## 1 0 17 11.64 11.11

## 5 7 NA 4.58

If **sorting** by frequency is not what I want I can easily turn it off by setting the sort parameter to FALSE:

ti\_tab1(

x = round(airquality$Solar.R, -2),

sort = FALSE

)

## value count pct pct\_all

## 1 0 17 11.64 11.11

## 2 100 34 23.29 22.22

## 3 200 50 34.25 32.68

## 4 300 45 30.82 29.41

## 5 7 NA 4.58

The same is true for the numbers of **digits** to show for the percentage columns:

ti\_tab1(

x = round(airquality$Solar.R, -2),

digits = 0

)

## value count pct pct\_all

## 3 200 50 34 33

## 4 300 45 31 29

## 2 100 34 23 22

## 1 0 17 12 11

## 5 7 NA 5

ti\_tab1(

x = round(airquality$Solar.R, -2),

digits = 4

)

## value count pct pct\_all

## 3 200 50 34.2466 32.6797

## 4 300 45 30.8219 29.4118

## 2 100 34 23.2877 22.2222

## 1 0 17 11.6438 11.1111

## 5 7 NA 4.5752

Since ti\_tab1() is implemented as generic it can handle **multiple data types** –  
i.e. vectors, data.frames, and lists – and can be extended to cover other data types as well.

Again the ti\_tab1() returns a data.frame. This time a column named  
name has been added which captures the name of the column on which the  
frequencies and percentages are based upon.

ti\_tab1(

x = lapply(airquality, round, -2)

)

## name value count pct pct\_all

## 1 Ozone 0 82 70.69 53.59

## 2 Ozone 37 NA 24.18

## 3 Ozone 100 33 28.45 21.57

## 4 Ozone 200 1 0.86 0.65

## 5 Solar.R 200 50 34.25 32.68

## 6 Solar.R 300 45 30.82 29.41

## 7 Solar.R 100 34 23.29 22.22

## 8 Solar.R 0 17 11.64 11.11

## 9 Solar.R 7 NA 4.58

## 10 Wind 0 153 100.00 100.00

## 11 Wind 0 NA 0.00

## 12 Temp 100 153 100.00 100.00

## 13 Temp 0 NA 0.00

## 14 Month 0 153 100.00 100.00

## 15 Month 0 NA 0.00

## 16 Day 0 153 100.00 100.00

## 17 Day 0 NA 0.00

Last but not least the fact that ti\_tab1() returns simple data.frames means  
that R provides a large array of things I can do with them – plotting, filtering, writing to file – and that every R user instantly knows how to handle them.

# get all counts

ti\_tab1(x = airquality$Wind)$count

## [1] 15 11 11 11 10 9 8 8 8 8 8 6 5 4 4 3 3 3 3 3 2 1 1 1 1 1 1 1 1 1 1 0

# get the highest percentage

tab <- ti\_tab1(x = round(airquality$Solar.R, -2))

tab$pct[1]

## [1] 34.25

# get percentage of NAs

tab$pct\_all[[is.na](http://is.na)(tab$value)]

## [1] 4.58

**Things to come**

As mentioned beforehand one of the things planed for this micro package is to add multidimensional tables. Another option is to extend the tabulation functions to allow for user defined aggregation functions producing other statistics than counts and percentages.

Other than that I think the package really has quite a narrow scope and we should keep it like that.