ascii

COLLABORATORS

	TITLE : ascii	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	David Hajage	April 3, 2009	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

List of Tables

ascii is a R package for writing asciidoc or txt2tags document with embeded R commands.

1 short example

```
<<>>=
x <- matrix(1:4, 2, 2)
x
@
```

gives :

```
> x <- matrix(1:4, 2, 2)
> x
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
<<results=ascii,echo=FALSE>>=
ascii(x, caption = "A simple matrix", width = 30)
@
```

gives :

1.00	3.00
2.00	4.00

Table 1: A simple matrix

2 what ascii provides

ascii provided :

- a generic method for common R objects: `ascii()`. Default argument depends of R object.
- two Sweave drivers: `Sweave("yourfile.Rnw", RweaveAsciidoc())` or `Sweave("yourfile.Rnw", RweaveT2t())`.

3 ascii manual

4 ascii examples

ascii provides methods for:

```
> methods(ascii)
 [1] ascii.anova*           ascii.aov*             ascii.aovlist*
 [4] ascii.cast_df*         ascii.character*       ascii.coxph*
 [7] ascii.data.frame*      ascii.default*         ascii.density*
[10] ascii.describe*        ascii.describe.single* ascii.factor*
[13] ascii.glm*             ascii.htest*           ascii.integer*
[16] ascii.list*            ascii.lm*              ascii.matrix*
```

x	An R object of class found among `methods(ascii)`.
include.rownames	logical. If `TRUE` the rows names are printed. Default value depends of class of `x`.
include.colnames	logical. If `TRUE` the columns names are printed. Default value depends of class of `x`.
format	Character vector of length equal to the number of columns of the resulting table (otherwise it will be replicated or truncated as necessary) indicating the format for the corresponding columns. These values are passed to the `formatC` function. Use `"d"` (for integers), `"f"`, `"e"`, `"E"`, `"g"`, `"G"`, `"fg"` (for reals), or `"s"` (for strings). `"f"` gives numbers in the usual `xxx.xxx` format; `"e"` and `"E"` give `n.ddde+nn` or `n.dddE+nn` (scientific format); `"g"` and `"G"` put `x[i]` into scientific format only if it saves space to do so. `"fg"` uses fixed format as `"f"`, but `digits` as number of *significant* digits. Note that this can lead to quite long result strings. Default depends on the class of `x`.
digits	Numeric vector of length equal to the number of columns of the resulting table (otherwise it will be replicated or truncated as necessary) indicating the number of digits to display in the corresponding columns. Default is `2`. decimal.mark: The character to be used to indicate the numeric decimal point. Default is `"."`.
na.print	The character string specifying how `NA` should be formatted specially. Default is `""`.
caption	Character vector of length 1 containing the table+s caption or title. Set to `""` to suppress the caption. Default value is `""`.
width	Numeric vector of length one containing the table width relative to the available width (expressed as a percentage value, `1... 99`). Default is `0` (all available width).
frame	Character vector of length one. Defines the table border, and can take the following values: `"topbot"` (top and bottom), `"all"` (all sides), `"none"` and `"sides"` (left and right). The default value is `""`.
grid	Character vector of length one. Defines which ruler lines are drawn between table rows and columns, and can take the following values: `"all"`, `"rows"`, `"cols"` and `"none"`. Default is `""`.
valign	Character vector of length one indicating vertical alignment of all cells in table. Can take the following values: `"top"`, `"bottom"` and `"middle"`. Default is `""`.
header	logical. If `TRUE` the first line of the table is emphasized. The default value depends of class of `x`.
footer	logical. If `TRUE` the last line of the table is emphasized. The default value depends of class of `x`.
align	Character vector of length one indicating the alignment of the corresponding columns. Can be composed with `"r"` (right), `"l"` (left) and `"c"` (center). Default value is `""`.
col.width	Numeric vector of length equal to the number of columns of the resulting table (otherwise it will be replicated or truncated as necessary) indicating width of the corresponding columns (integer proportional values). Default is `1`.
style	Character vector of length one indicating the style of the corresponding columns. Can be composed with `"d"` (default), `"e"` (emphasis), `"m"` (monospaced), `"a"` (cells can contain any of the AsciiDoc elements that are allowed inside document), `"l"` (literal), `"v"` (verse; all line breaks are retained). Default is `""`.
...	Additional arguments. (Currently ignored.)

Table 2: ascii

x	An object of class "ascii"
type	Type of syntax produce. Possible values for type are "asciidoc", "t2t" or "textile". Default value produce AsciiDoc syntax.
list.type	Character vector of length one indicating the list type ("bullet", "number" or "none"). Default is *"bullet"*.
...	Additional arguments. (Currently ignored.)

Table 3: print.ascii

```
[19] ascii.numeric*         ascii.prcomp*          ascii.smooth.spline*
[22] ascii.summary.aov*     ascii.summary.aovlist* ascii.summary.glm*
[25] ascii.summary.lm*      ascii.summary.prcomp*  ascii.summary.table*
[28] ascii.survdiff*        ascii.tab1*            ascii.table*
[31] ascii.ts*              ascii.zoo*

   Non-visible functions are asterisked
```

4.1 vector

```
> ascii(1:4)
|==================
|1.00|2.00|3.00|4.00
|==================
```

1.00	2.00	3.00	4.00

4.2 matrix

```
> ascii(VADeaths, include.rownames = T, include.colnames = T, caption = "VADeaths",
+     header = T, col.width = c(1, 2, 2, 2, 2, 2), valign = "middle",
+     align = "lrrrr", frame = "topbot")
.VADeaths
[frame="topbot",valign="middle",options="header",cols="<1,>2,>2,>2,>2"]
|=================================================
|     |Rural Male|Rural Female|Urban Male|Urban Female
|50-54|11.70     |8.70        |15.40     |8.40
|55-59|18.10     |11.70       |24.30     |13.60
|60-64|26.90     |20.30       |37.00     |19.30
|65-69|41.00     |30.90       |54.60     |35.10
|70-74|66.00     |54.30       |71.10     |50.00
|=================================================
```

	Rural Male	Rural Female	Urban Male	Urban Female
50-54	11.70	8.70	15.40	8.40
55-59	18.10	11.70	24.30	13.60
60-64	26.90	20.30	37.00	19.30
65-69	41.00	30.90	54.60	35.10
70-74	66.00	54.30	71.10	50.00

Table 4: VADeaths

4.3 data.frame

```
> ascii(iris[1:10, ], include.rownames = F, caption = "iris", width = 75,
+     align = "c", valign = "bottom", frame = "topbot", grid = "none")
.iris
[frame="topbot",grid="none",valign="bottom",options="header",cols="^,^,^,^,^",width="75%"]
```

```
|5.00         |3.60         |1.40         |0.20         |setosa
|5.40         |3.90         |1.70         |0.40         |setosa
|4.60         |3.40         |1.40         |0.30         |setosa
|5.00         |3.40         |1.50         |0.20         |setosa
|4.40         |2.90         |1.40         |0.20         |setosa
|4.90         |3.10         |1.50         |0.10         |setosa
|=========================================================
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.10	3.50	1.40	0.20	setosa
4.90	3.00	1.40	0.20	setosa
4.70	3.20	1.30	0.20	setosa
4.60	3.10	1.50	0.20	setosa
5.00	3.60	1.40	0.20	setosa
5.40	3.90	1.70	0.40	setosa
4.60	3.40	1.40	0.30	setosa
5.00	3.40	1.50	0.20	setosa
4.40	2.90	1.40	0.20	setosa
4.90	3.10	1.50	0.10	setosa

Table 5: iris

4.4 summary.table

```
> ascii(summary(table(1:4, 1:4)))
* Number of cases in table: 4
* Number of factors: 2
* Test for independence of all factors:
** Chisq = 12, df = 9, p-value = 0.2133
** Chi-squared approximation may be incorrect
```

- Number of cases in table: 4
- Number of factors: 2
- Test for independence of all factors:
  - Chisq = 12, df = 9, p-value = 0.2133
  - Chi-squared approximation may be incorrect

4.5 glm

```
> counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
> outcome <- gl(3, 1, 9)
> treatment <- gl(3, 3)
> d.AD <- data.frame(treatment, outcome, counts)
> glm.D93 <- glm(counts ~ outcome + treatment, family = poisson())
> glm.D93
Call:  glm(formula = counts ~ outcome + treatment, family = poisson())

Coefficients:
(Intercept)     outcome2     outcome3   treatment2   treatment3
  3.045e+00   -4.543e-01   -2.930e-01    8.717e-16    4.557e-16

Degrees of Freedom: 8 Total (i.e. Null);  4 Residual
```

```
Null Deviance:      10.58
Residual Deviance: 5.129        AIC: 56.76
> ascii(glm.D93, caption = "glm.D93")
.glm.D93
[options="header"]
|================================================
|           |Estimate|Std. Error|z value|Pr(>\|z\|)
|(Intercept)|3.04    |0.17      |17.81  |0.00
|outcome2   |-0.45   |0.20      |-2.25  |0.02
|outcome3   |-0.29   |0.19      |-1.52  |0.13
|treatment2 |0.00    |0.20      |0.00   |1.00
|treatment3 |0.00    |0.20      |0.00   |1.00
|================================================
> ascii(anova(glm.D93), caption = "anova glm.D93", include.rownames = T)
.anova glm.D93
[options="header"]
|=========================================
|          |Df  |Deviance|Resid. Df|Resid. Dev
|NULL      |    |        |8.00     |10.58
|outcome   |2.00|5.45    |6.00     |5.13
|treatment|2.00|0.00    |4.00     |5.13
|=========================================
```

	Estimate	**Std. Error**	**z value**	**Pr(>|z|)**
(Intercept)	3.04	0.17	17.81	0.00
outcome2	-0.45	0.20	-2.25	0.02
outcome3	-0.29	0.19	-1.52	0.13
treatment2	0.00	0.20	0.00	1.00
treatment3	0.00	0.20	0.00	1.00

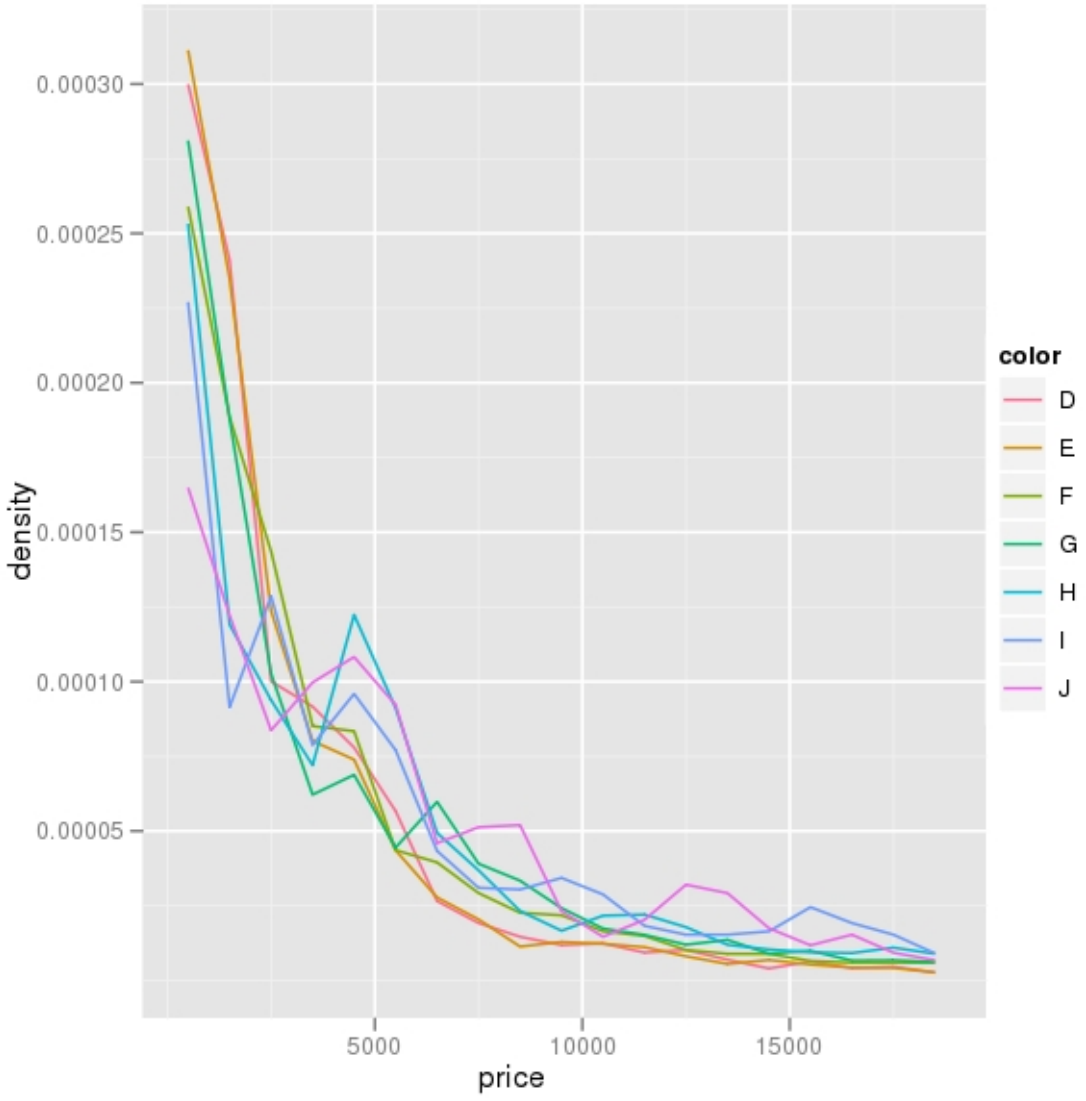
Table 6: glm.D93

	Df	**Deviance**	**Resid. Df**	**Resid. Dev**
NULL			8.00	10.58
outcome	2.00	5.45	6.00	5.13
treatment	2.00	0.00	4.00	5.13

Table 7: anova glm.D93

4.6 plot

```
> library(ggplot2)
> p <- qplot(price, ..density.., data = diamonds, geom = "freqpoly",
+     binwidth = 1000, colour = color)
> print(p)
```



4.7 txt2tags

```
> library(reshape)
> names(airquality) <- tolower(names(airquality))
> aqm <- melt(airquality, id = c("month", "day"), na.rm = TRUE)
> res <- cast(aqm, month ~ variable, mean, margins = "grand_row")
> res
  month    ozone  solar.r      wind     temp
1     5 23.61538 181.2963 11.622581 65.54839
2     6 29.44444 190.1667 10.266667 79.10000
3     7 59.11538 216.4839  8.941935 83.90323
4     8 59.96154 171.8571  8.793548 83.96774
5     9 31.44828 167.4333 10.180000 76.90000
6 (all) 42.12931 185.9315  9.957516 77.88235
> print(ascii(res), "t2t")
|| month | ozone | solar.r | wind  | temp  |
| 5     | 23.62 | 181.30  | 11.62 | 65.55 |
| 6     | 29.44 | 190.17  | 10.27 | 79.10 |
| 7     | 59.12 | 216.48  | 8.94  | 83.90 |
| 8     | 59.96 | 171.86  | 8.79  | 83.97 |
| 9     | 31.45 | 167.43  | 10.18 | 76.90 |
```

```
| (all) | 42.13 | 185.93  | 9.96  | 77.88 |
```

5 convert

`Sweave` process creates a `yourdocument.txt` file from `yourdocument.Rnw`.

```
Sweave("youdocument.Rnw", RweaveXxx)
```

You can convert it to html format with the following command:

```
asciidoc yourdocument.txt
or
txt2tags -t html yourdocument.txt
```

or to xhtml, docbook, man, tex...

For example, you can see the source of this documentation, the file generated by Sweave, the same file in docbook format, the same file converted to pdf with dblatex, and the same file converted to odt with docbook2odf.