Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

xkcd: An R Package for Plotting XKCD Graphs

Emilio Torres-Manzanera
University of Oviedo

Abstract

XKCD is a popular stick figure web comic with themes in mathematics, science, language, and romance created by Randall Munroe. Folk have figured out how to make XKCD style graphs in Mathematica, in python, LaTeX and in other programming languages. But so far there is no an elegant solution to make it in R. **xkcd** gives a satisfactory answer to the question *How can we make XKCD style graphs in R?*. It provides a set of functions for plotting data in an XKCD style using **ggplot2**.

Keywords: XKCD, stick figure, **ggplot2**, **xkcd**, R.

1. Introduction and Main Motivation

The XKCD web site is a web comic created by Munroe (2005), who described it as a web comic of romance, sarcasm, math, and language. He use stick figures, very simple drawings composed of a few lines and curves, in his comics strips, that are available under the Creative Commons Attribution-NonCommercial 2.5 License (Fig. 1).

Due to the popularity of Randall Munroe's work, several questions about how to draw XKCD style graphs using different programming languages were posted in Stack Overflow, a language-independent collaboratively edited question and answer site for programmers. Many authors tried to replicate such style using different programming languages. For instance, Woods (2012) and Layton (2012) created functions that apply a distortion using image processing to existing charts in Mathematica and in Matlab, respectively; there exists a python library focused on this style (Hunter 2012); or percusse (2012) proposed an example in LaTeX (Fig. 2).

The R language (R Core Team 2014), despite of being a software environment for statistical computing and graphics, lacked of a specific package devoted to plot graphs in an XKCD style. **RXKCD** (Sonego 2012) allowed the visualisation of XKCD comic strip directly from R, and there was just a function to plot a tree with this style in **phytools** (Revell 2012).

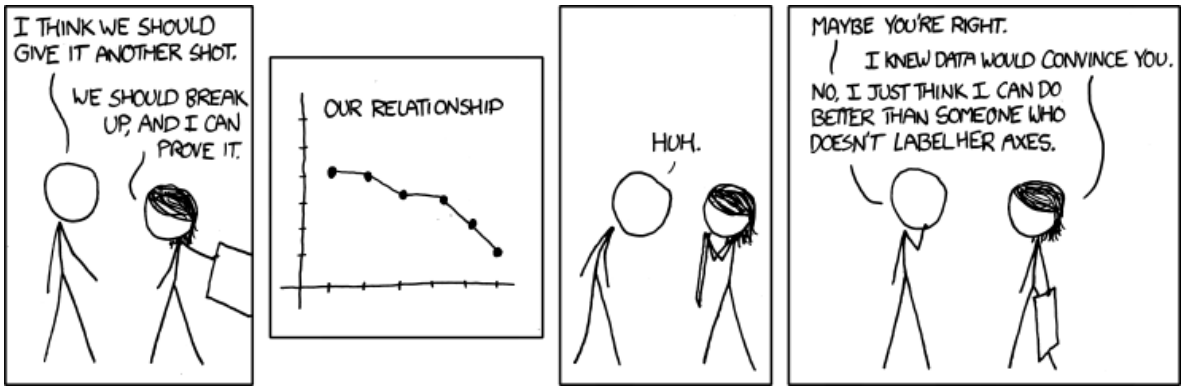
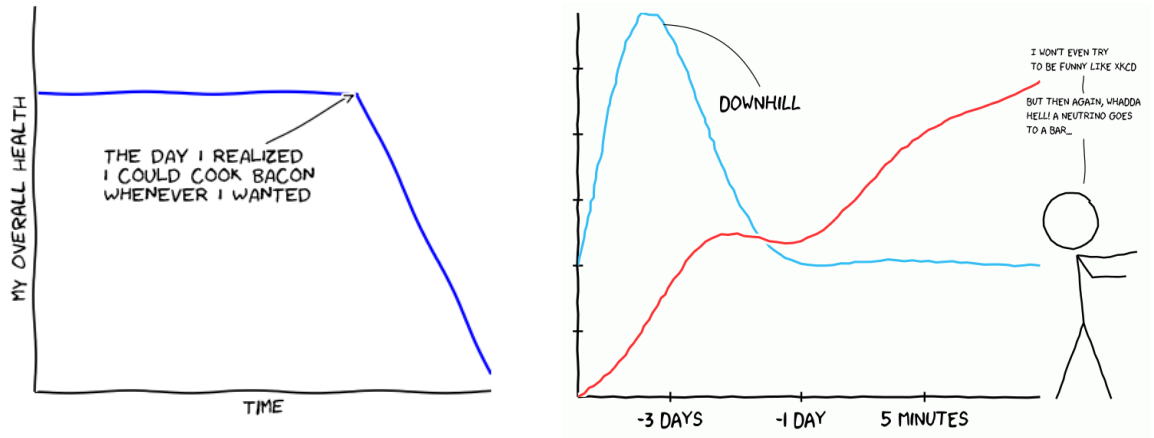


Figure 1: Convincing (And if you labeled your axes, I could tell you exactly how much better) (Munroe 2011b).



(a) Python (b) LaTeX

Figure 2: Examples of XKCD graphs in several programming languages.
Source: Hunter (2012); percusse (2012)

To the best of our knowledge, there was not a satisfactory response to the question *How can we make xkcd style graphs in R?* published in Stack Overflow by jebyrnes (2012). There was not a specific package devoted to draw stick figures, where the head is represented by a circle an the arms, legs and torso are all represented by straight lines, nor to plot axis or lines as in a handmade drawing.

Fortunately, R is rich with facilities for creating and developing interesting graphics (Lewin-Koh 2013). For instance, **lattice** (Sarkar 2008) and **ggplot2** (Wickham 2009) allowed for building many types of plots with sophisticated layouts. In particular, the implementation of the semantic for graphics of Wilkinson (2005) in R by Wickham (2009) allowed the design of XKCD style graphs. The package **xkcd** presented in this paper develops several functions to plot statistical graphs in a freehand sketch style following the guidelines of the XKCD web site.

In this paper, an introduction to plot such graphs in R is detailed. The following section deals with installing xkcd fonts and saving graphs, a step that raised several questions among users. In the next section some basic examples are explained and the graph gallery section shows several complex plots. Finally, the article ends with a discussion concerning statistical information and creative drawing.

2. The XKCD fonts

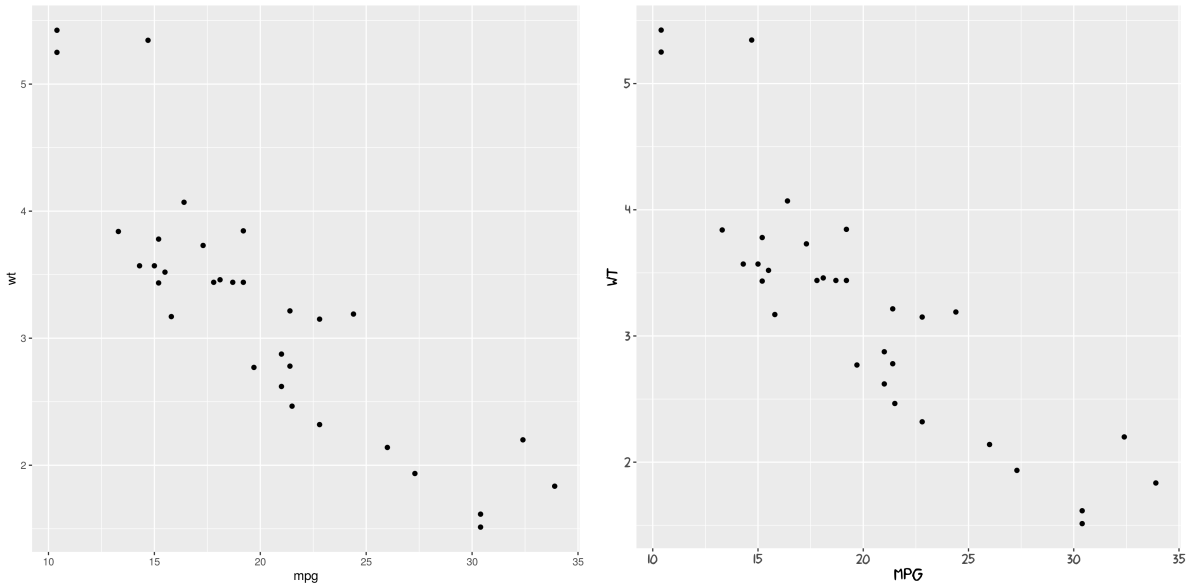
The package **xkcd** uses the XKCD fonts, so you must install them in your computer. An easy way to check whether this fonts are installed in the computer or not is typing the following code and comparing the results (Fig. 3):

```
> library(extrafont)
> library(ggplot2)
> if( 'xkcd' %in% fonts()) {
+   p <-  ggplot() + geom_point(aes(x=mpg, y=wt), data=mtcars) +
+     theme(text = element_text(size = 16, family = "xkcd"))
+ } else  {
+   warning("Not xkcd fonts installed!")
+   p <-  ggplot() + geom_point(aes(x=mpg, y=wt), data=mtcars)
+ }
> p
```

Look at the labels and check the type of font that appears on them. In the case that you do not see the XKCD fonts, you should get and install them following the instructions of the next subsection.

2.1. Installing the xkcd fonts in the computer

If the XKCD fonts are not installed in the system, you should install them using the package **extrafont** developed by Chang (2014). In particular, the function `font_import()` registers font files and assigns a family name, `fonts()` lists available type fonts, `fonttable()` lists available font families, and finally, `loadfonts()` registers the fonts in the Adobe font metric table with R's PDF (portable document format) or PostScript output device.



(a) No XKCD fonts (b) XKCD fonts installed

Figure 3: Are XKCD fonts installed on your computer?

```
> library(extrafont)
> download.file("http://simonsoftware.se/other/xkcd.ttf",
+               dest="xkcd.ttf", mode="wb")
> system("mkdir ~/.fonts")
> system("cp xkcd.ttf  ~/.fonts")
> font_import(pattern = "[X/x]kcd", prompt=FALSE)
> fonts()
> fonttable()
> if(.Platform$OS.type != "unix") {
+   ## Register fonts for Windows bitmap output
+   loadfonts(device="win")
+ } else {
+   loadfonts()
+ }
```

In the previous code, a font file in format `ttf` (TrueType outline font) is downloaded from a external site and installed in the computer. If you want to modify these fonts, you may use the spline font database file format (`sfd`), the official xkcd font created by Kelley (2013) and distributed under a Creative Commons Attribution-NonCommercial 3.0 License, which can be opened with FontForge, an open-source font editor (Williams 2012).

2.2. Saving the graphs

`ggsave()` is the preferred function for saving a **ggplot2** plot. For instance, the following instruction saves the graph as a bitmap file:

```
> ggsave("gr1.png", p)
```

If you want to save this chart as PDF you should embed the fonts into the PDF file, using `embed_fonts()` (Chang 2014). First, if you are running Windows, you may need to tell it where the Ghostscript program is, for embedding fonts.

```
> ggsave("gr1.pdf", plot=p,  width=12, height=4)
> if(.Platform$OS.type != "unix") {
+    ## Needed for Windows. Make sure you have the correct path
+    Sys.setenv(R_GSCMD =
+                "C:\\Program Files (x86)\\gs\\gs9.06\\bin\\gswin32c.exe")
+ }
> embed_fonts("gr1.pdf")
```

See the development site of **extrafont** for additional instructions and examples of using fonts other than the standard PostScript fonts, like TrueType fonts, with PDF or PostScript output files, and with bitmap output files in Windows (Chang 2014).

3. Installing xkcd

The home site of **xkcd** is located at R-forge, a central platform for the development of R packages. From within R, you can install the stable version from CRAN (The Comprehensive R Archive Network) with the following instruction:

```
> install.packages("xkcd",dependencies = TRUE)
```

Then, you may want to see the vignette and check the code:

```
> help(package="xkcd")
> vignette("xkcd-intro") # It opens the PDF
> browseVignettes(package = "xkcd") # To browse the PDF, R and Rnw
```

Once the package has been installed, it can be loaded by typing:

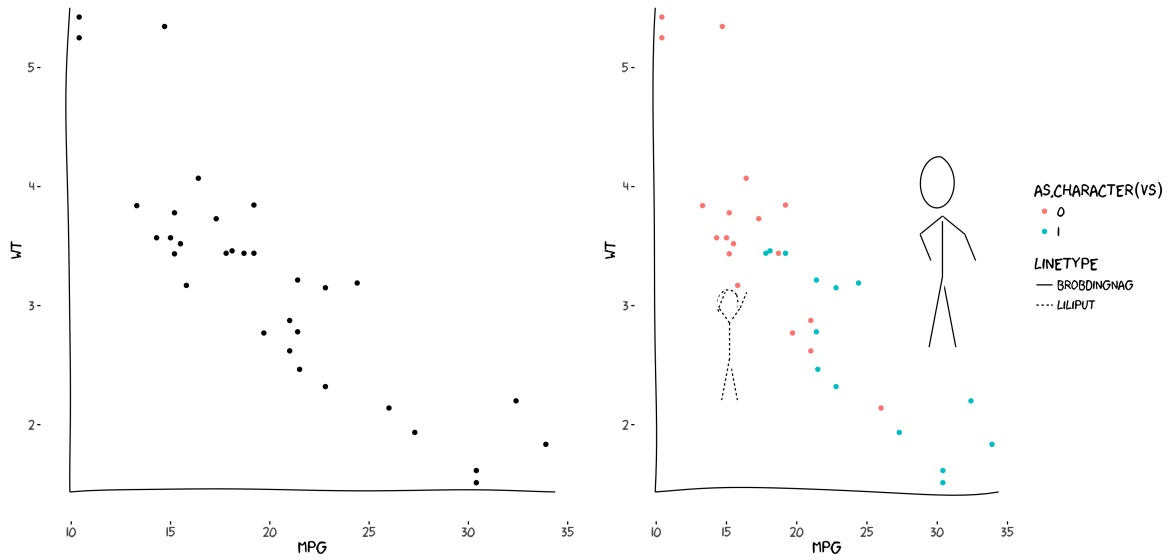
```
> library(xkcd)
```

Automatically, it loads the packages **ggplot2** and **extrafont**. The most relevant functions are `xkcdaxis()`, that plots axis in a handwritten style, `xkcdman()`, that draws stick figures, and `xkcdrect()`, that creates fuzzy rectangles. These functions are compatible with the grammar of graphics proposed by Wickham (2009).

From a technical point of view, lines plotted with the **xkcd** package are jittered with white noise and then smoothed using curves of Bézier with the `bezier()` function from **Hmisc** (Frank E Harrell Jr and with contributions from Charles Dupont and many others 2014).

4. Axis, Stick Figures and Facets

In order to get a handmade drawing of the axis and an XKCD theme, a specific function based on the **ggplot2** framework was created, `xkcdaxis()` (Fig. 4a):



(a) xkcd theme (b) Stick figures

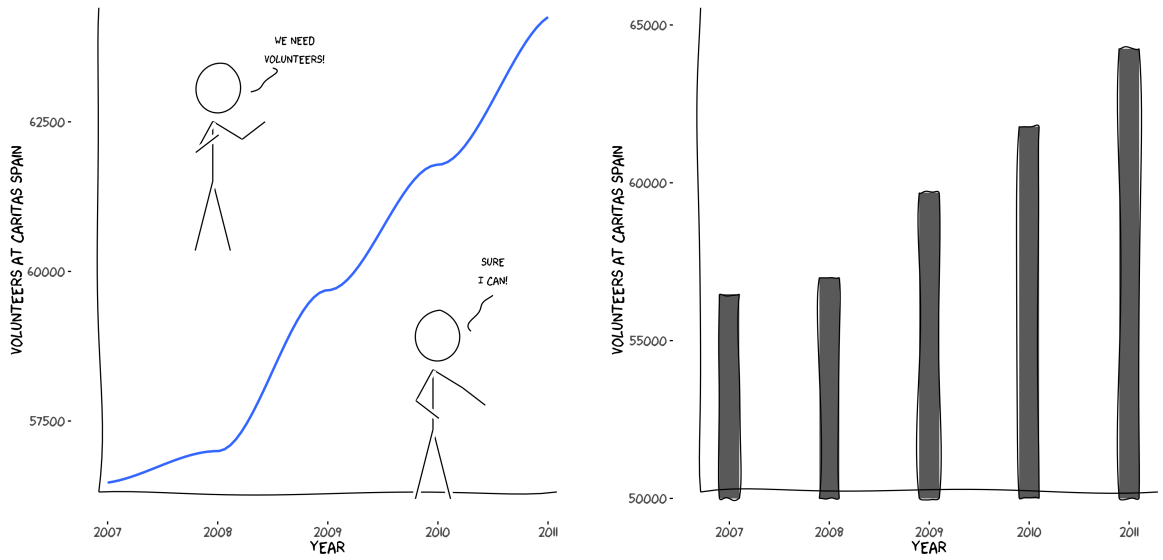
Figure 4: Creating handmade axis and stickmen.

```
> xrange <- range(mtcars$mpg)
> yrange <- range(mtcars$wt)
> set.seed(123) # for reproducibility
> p <- ggplot() + geom_point(aes(mpg, wt), data=mtcars) +
+       xkcdaxis(xrange,yrange)
> p
```

Due to the fact that white random noise is used in the picture, it is convenient to fix a seed for reproducing the same figure (`set.seed()`).

May be the most flashy function is `xkcdman()`, that draws a stick figure with different positions, angles and lengths (Fig. 4b).

```
> ratioxy <- diff(xrange)/diff(yrange)
> mapping <- aes(x, y, scale, ratioxy, angleofspine,
+                anglerighthumerus, anglelefthumerus,
+                anglerightradius, angleleftradius,
+                anglerightleg, angleleftleg, angleofneck,
+                linetype=city)
> dataman <- data.frame(x= c(15,30), y=c(3, 4),
+                       scale = c(0.3,0.51) ,
+                       ratioxy = ratioxy,
+                       angleofspine =  -pi/2  ,
+                       anglerighthumerus = c(pi/4, -pi/6),
+                       anglelefthumerus = c(pi/2 + pi/4, pi +pi/6),
+                       anglerightradius = c(pi/3, -pi/3),
+                       angleleftradius = c(pi/3, -pi/3),
```



(a) Line graph (b) Bar graph

Figure 5: Some basic examples

```
+                       anglerightleg = 3*pi/2  - pi / 12,
+                       angleleftleg = 3*pi/2  + pi / 12 ,
+                       angleofneck = runif(1, 3*pi/2-pi/10, 3*pi/2+pi/10),
+                       city=c("Liliput","Brobdingnag"))
> p <- ggplot() + geom_point(aes(mpg, wt, colour=as.character(vs)), data=mtcars) +
+   xkcdaxis(xrange,yrange) +
+   xkcdman(mapping, dataman)
> p
```

Additionally, you may use the facet option of **ggplot2** to do split up your data by one or more variables and plot the subsets of data together.

```
> p +  facet_grid(.~vs)
```

5. Some Basic Examples

In this section the code to plot a line graph and a bar chart is detailed (Fig. 5). Remember to set the seed if you want to replicate your own figures (see Section 4).

```
> volunteers <- data.frame(year=c(2007:2011),
+                          number=c(56470, 56998, 59686, 61783, 64251))
> xrange <- range(volunteers$year)
> yrange <- range(volunteers$number)
> ratioxy <-  diff(xrange) / diff(yrange)
> datalines <- data.frame(xbegin=c(2008.3,2010.5),ybegin=c(63000,59600),
```

```
+                          xend=c(2008.5,2010.3), yend=c(63400,59000))
> mapping <- aes(x, y, scale, ratioxy, angleofspine,
+                anglerighthumerus, anglelefthumerus,
+                anglerightradius, angleleftradius,
+                anglerightleg, angleleftleg, angleofneck)
> dataman <- data.frame( x= c(2008,2010), y=c(63000, 58850),
+                        scale = 1000 ,
+                        ratioxy = ratioxy,
+                        angleofspine =  -pi/2  ,
+                        anglerighthumerus = c(-pi/6, -pi/6),
+                        anglelefthumerus = c(-pi/2 - pi/6, -pi/2 - pi/6),
+                        anglerightradius = c(pi/5, -pi/5),
+                        angleleftradius = c(pi/5, -pi/5),
+                        angleleftleg = 3*pi/2  + pi / 12 ,
+                        anglerightleg = 3*pi/2  - pi / 12,
+                        angleofneck = runif(1, 3*pi/2-pi/10, 3*pi/2+pi/10))
> p <- ggplot() + geom_smooth(mapping=aes(x=year, y =number),
+                             data =volunteers, method="loess") +
+   xkcdaxis(xrange,yrange) +
+   ylab("Volunteers at Caritas Spain") +
+   xkcdman(mapping, dataman) +
+   annotate("text", x=2008.7, y = 63700,
+            label = "We Need\nVolunteers!", family="xkcd" ) +
+   annotate("text", x=2010.5, y = 60000,
+            label = "Sure\nI can!", family="xkcd" ) +
+   xkcdline(aes(xbegin=xbegin,ybegin=ybegin,xend=xend,yend=yend),
+            datalines, xjitteramount = 0.12)
> p # Figure 5.a
```

One interesting property of this charts is that when xkcd lines intersect, there is a space in blank in the crossing, following XKCD style (Fig. 5a).

The following code plots the line char as a bar chart (Fig. 5b):

```
> data <- volunteers
> data$xmin <- data$year - 0.1
> data$xmax <- data$year + 0.1
> data$ymin <- 50000
> data$ymax <- data$number
> xrange <- range(min(data$xmin)-0.1, max(data$xmax) + 0.1)
> yrange <- range(min(data$ymin)+500, max(data$ymax) + 1000)
> mapping <- aes(xmin=xmin,ymin=ymin,xmax=xmax,ymax=ymax)
> p <- ggplot() + xkcdrect(mapping,data) +
+   xkcdaxis(xrange,yrange) +
+   xlab("Year") + ylab("Volunteers at Caritas Spain")
> p # Figure 5.b
```

You may specify the width, the height, and colours of the bars (see `example(xkcdrect)`).

6. Graph Gallery

This gallery provides a variety of complex charts designed to address your idea (Fig. 6). The code is in the `.Rnw` file of this paper and in the R-forge site, where there are more elaborate examples. Figures 6a and 6d fill the area under the data with letters, Figure 6b represents line graphs, and Figure 6c shows a population pyramidal.

7. Discussion

As Chen, Härdle, and Unwin (2007) stated, visualising the data is an essential part of any data analysis, but the role graphics played to enlighten the audience means different things to different people. On the one hand, statisticians argue that graphic display need to be clear, concise and accurate; on the other hand, artists say that to be effective, it need to be eye-catching, engaging, and innovative (Holmes, Peltier, and Robbin 2012).

As Munroe (2011a) points out, the book of Tufte (1983) is a reference in such field as the visual communication of information. Tufte (1983) coined the word *chartjunk* to refer to useless, non-informative, or information-obscuring elements of quantitative information displays and argue against using excessive decoration. Overloading your chart with XKCD elements may lead to loss of accuracy and the data visualisation might become a chartjunk.

In order to avoid this error, you must think about what the consumers of your graphs are looking for. Charts designed for business decisions and those designed for t-shirts should not be the same. If you are trying to provoke or entertain, the XKCD style graphs may be suitable for your purpose and meaningful to your audience.

In summary, the package **xkcd** is a **ggplot2** wrapper that makes graphs look like they came from an XKCD comic and it gives a satisfactory answer to the question *How can we make XKCD style graphs in R?*

References

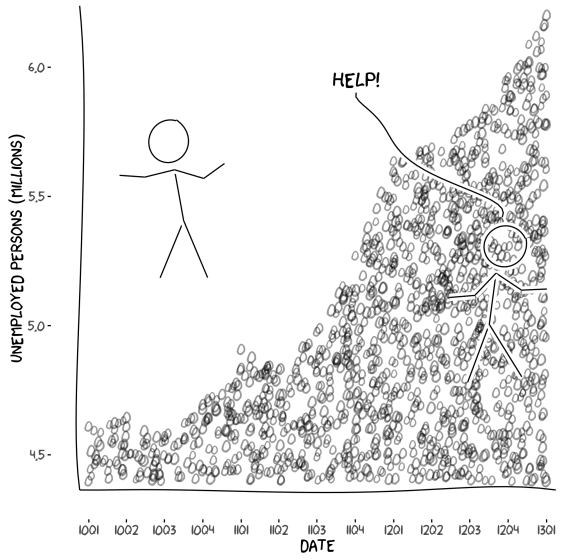
Chang W (2014). *extrafont: Tools for using fonts*. R package version 0.17, URL http://CRAN.R-project.org/package=extrafont.

Chen C, Härdle W, Unwin A (2007). *Handbook of Data Visualization*. Springer Handbooks of Computational Statistics. Springer. ISBN 9783540330370. URL http://books.google.es/books?id=zzCiSJoohuQC.

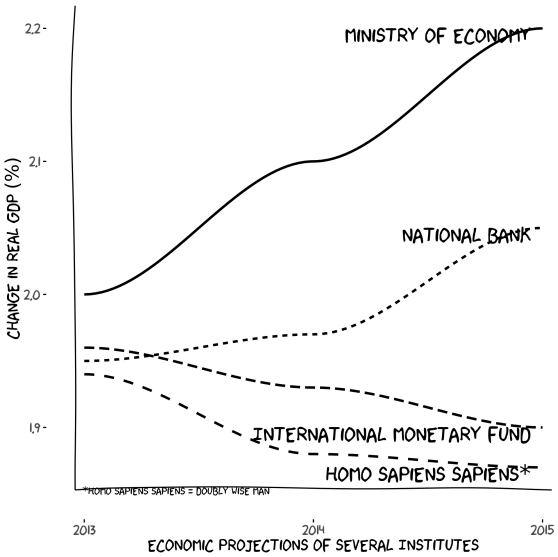
Frank E Harrell Jr and with contributions from Charles Dupont and many others (2014). *Hmisc: Harrell Miscellaneous*. R package version 3.14-4, URL http://CRAN.R-project.org/package=Hmisc.

Holmes N, Peltier J, Robbin N (2012). "Data Visualization: Statistical Graphics or Data Art?" Last accessed on December 08, 2014, URL http://strataconf.com/stratany2012/public/schedule/detail/25088.

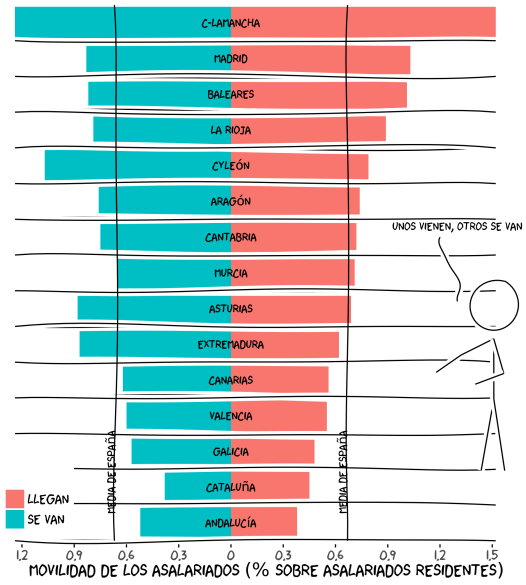
Hunter JD (2012). "Showcase example code: xkcd.py." Last accessed on December 08, 2014, URL http://matplotlib.org/xkcd/examples/showcase/xkcd.html.



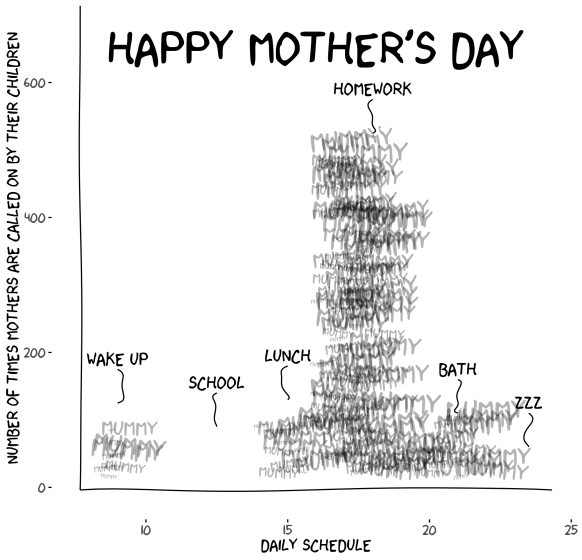
(a) Help!



(b) Homo Sapiens Sapiens



(c) A population pyramid



(d) Mother's day

Figure 6: Graph gallery

jebyrnes (2012). "How can we make xkcd style graphs in R?" Last accessed on December 08, 2014, URL http://stackoverflow.com/questions/12675147/how-can-we-make-xkcd-style-graphs-in-r.

Kelley K (2013). "xkcd-font." Last accessed on December 08, 2014, URL https://github.com/ipython/xkcd-font.

Layton S (2012). "XKCDIFY! Adding flair to boring Matlab Axes one plot at a time." Last accessed on December 08, 2014, URL https://github.com/slayton/matlab-xkcdify.

Lewin-Koh N (2013). "CRAN Task View: Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization." Last accessed on December 08, 2014, URL http://cran.r-project.org/web/views/Graphics.html.

Munroe R (2005). "XKCD, A webcomic of romance, sarcasm, math, and language." Last accessed on December 08, 2014, URL http://xkcd.com/.

Munroe R (2011a). "Blogofractal." Last accessed on December 08, 2014, URL http://www.xkcd.com/124/.

Munroe R (2011b). "Convincing (And if you labeled your axes, I could tell you exactly how MUCH better)." Last accessed on December 08, 2014, URL http://xkcd.com/833/.

percusse (2012). "Create xkcd style diagram in TeX." Last accessed on December 08, 2014, URL http://tex.stackexchange.com/questions/74878/create-xkcd-style-diagram-in-tex.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Revell LJ (2012). "phytools: An R package for phylogenetic comparative biology (and other things)." *Methods in Ecology and Evolution*, **3**, 217–223.

Sarkar D (2008). *Lattice: Multivariate Data Visualization with R*. Springer, New York. ISBN 978-0-387-75968-5, URL http://lmdvr.r-forge.r-project.org.

Sonego P (2012). *RXKCD: Get XKCD comic from R*. R package version 1.7-5, URL http://CRAN.R-project.org/package=RXKCD.

Tufte ER (1983). *The Visual Display of Quantitative Information*. Graphics Press, Cheshire.

Wickham H (2009). *ggplot2: elegant graphics for data analysis*. Springer New York. ISBN 978-0-387-98140-6. URL http://had.co.nz/ggplot2/book.

Wilkinson L (2005). *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 0387245448.

Williams G (2012). "FontForge." Last accessed on December 08, 2014, URL http://fontforge.github.io/en-US/.

Woods S (2012). "xkcd-style graphs." Last accessed on December 08, 2014, URL http://mathematica.stackexchange.com/questions/11350/xkcd-style-graphs/11355#11355.

Contents

Affiliation:

Emilio Torres-Manzanera
Department of Statistics
Faculty of Comerce
University of Oviedo
C/ Luis Moya Blanco, 261, 33203 Gijón, Spain
E-mail: torres@uniovi.es
URL: http://uce.uniovi.es/

Journal of Statistical Software https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd