XKCD

Bruce Campbell

August 8, 2017

XKCD Demo

There are four functions and a font mechanism involved in making XKCD graphics.

`xkcdaxis xkcdline xkcdman xkcdrect`

The font you want to install is xkcd.ttf

```
# install.packages('xkcd')

library(xkcd)

# vignette('xkcd-intro')


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)
```

```
## Scanning ttf files in C:\windows\Fonts ...
## Extracting .afm files from .ttf files...
## C:\Windows\Fonts\xkcd.ttf
##  : xkcd already registered in fonts database. Skipping.
## Found FontName for 0 fonts.
## Scanning afm files in C:/Program Files/R/R-3.2.4revised/library/extrafontdb/metrics
```

```
fonts()
```

```
## [1] "xkcd"
```

```
fonttable()
```

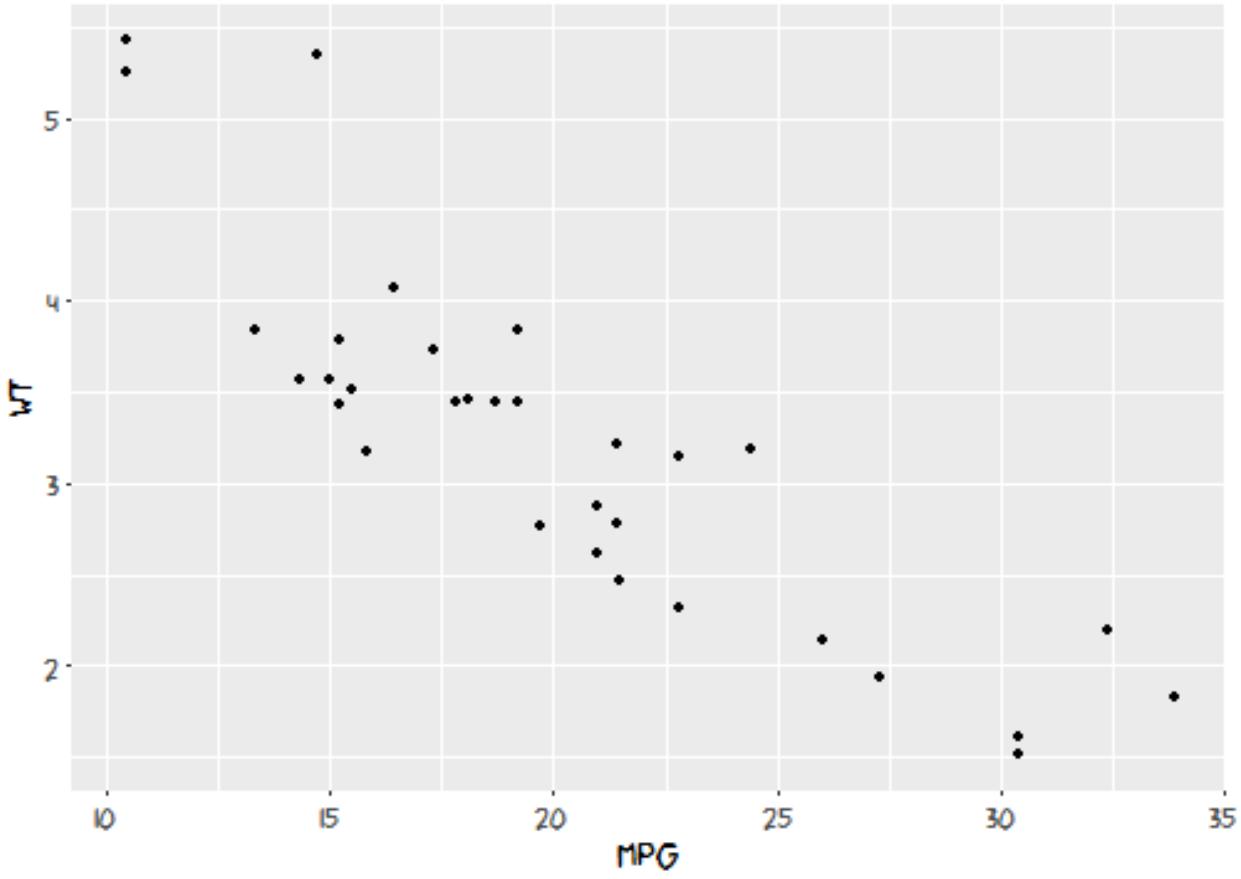
```
##   package     afmfile                     fontfile FullName FamilyName
## 1      NA xkcd.afm.gz C:\\Windows\\Fonts\\xkcd.ttf     xkcd       xkcd
##   FontName  Bold Italic Symbol afmsymfile
## 1     xkcd FALSE  FALSE  FALSE         NA
```

```
if (.Platform$OS.type != "unix") {
    ## Register fonts for Windows bitmap output
    loadfonts(device = "win")
} else {
    loadfonts()
}
```

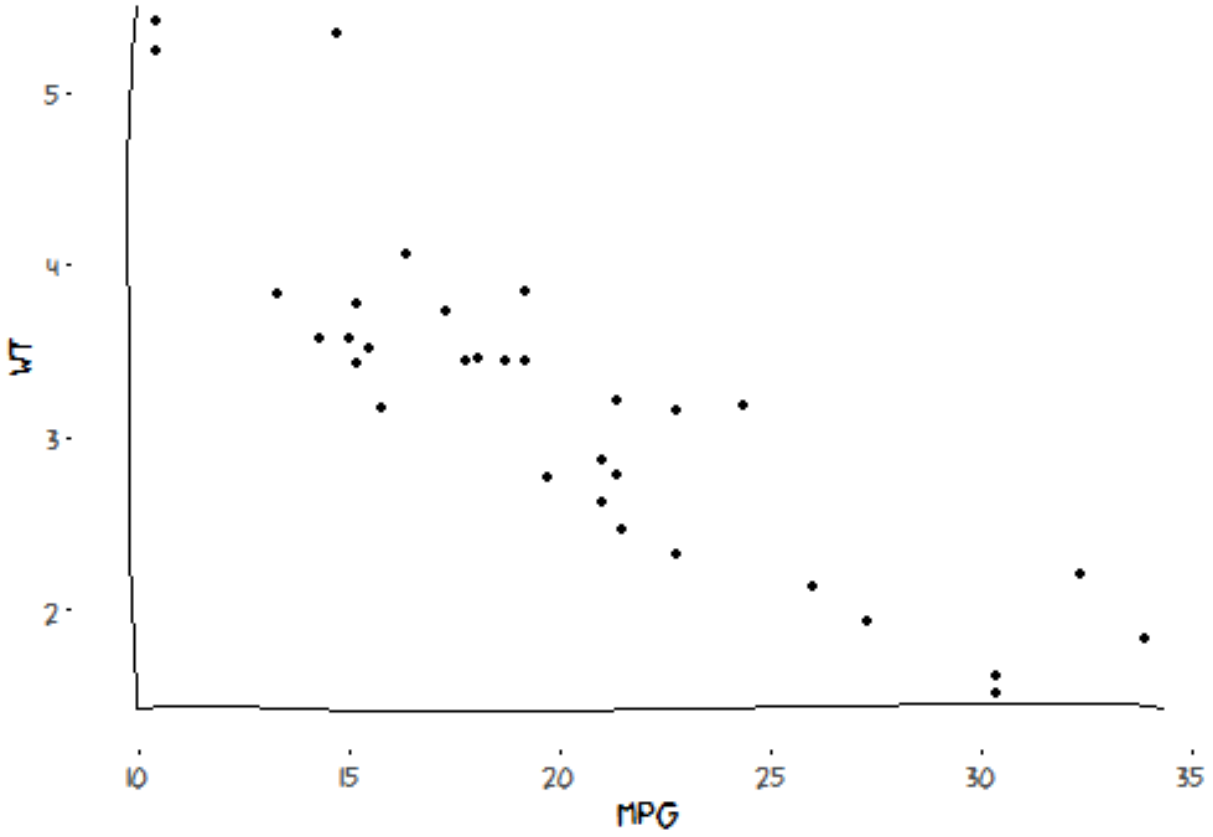
```
## xkcd already registered with windowsFonts().
```

```
library(extrafont)
library(ggplot2)
```

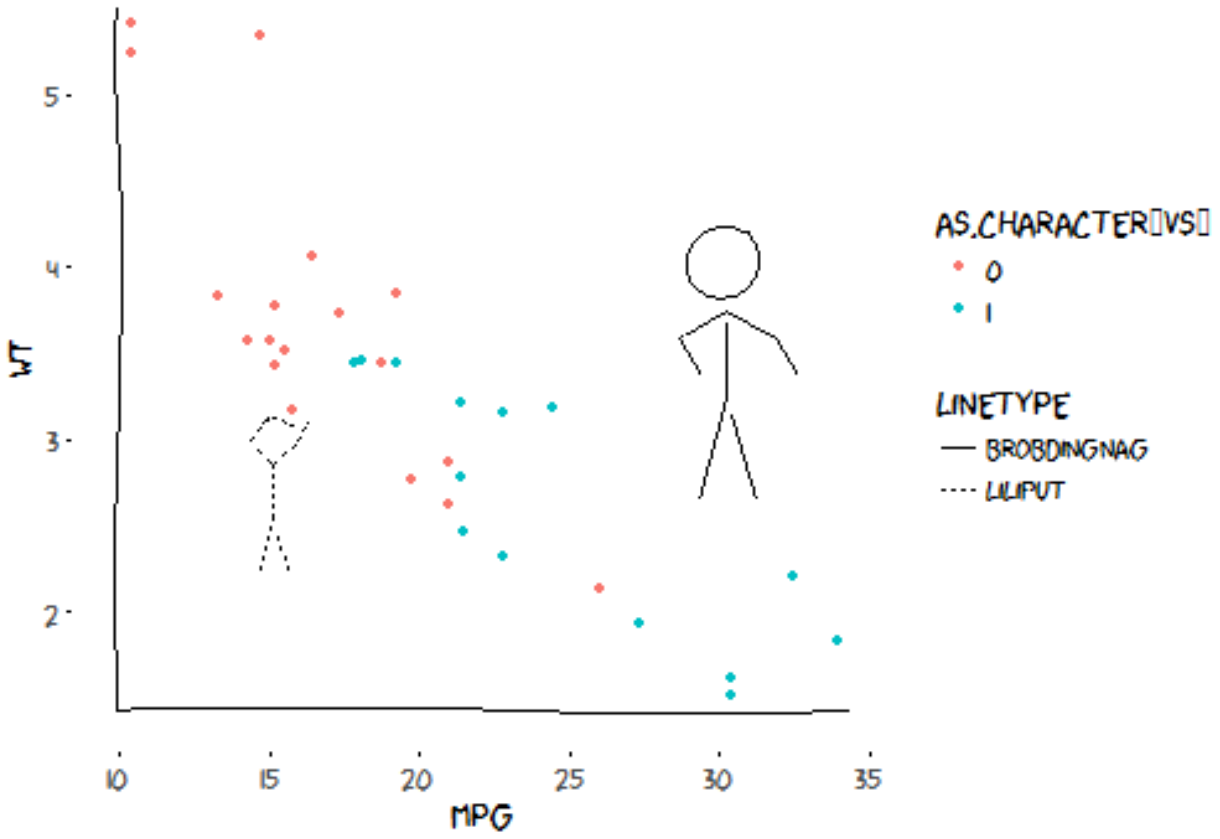
```
if ("xkcd" %in% fonts()) {
    p <- ggplot() + geom_point(aes(x = mpg, y = wt), data = mtcars) + theme(text = element_text(size = 
        family = "xkcd"))
} else {
    warning("Not xkcd fonts installed!")
    p <- ggplot() + geom_point(aes(x = mpg, y = wt), data = mtcars)
}
p
```



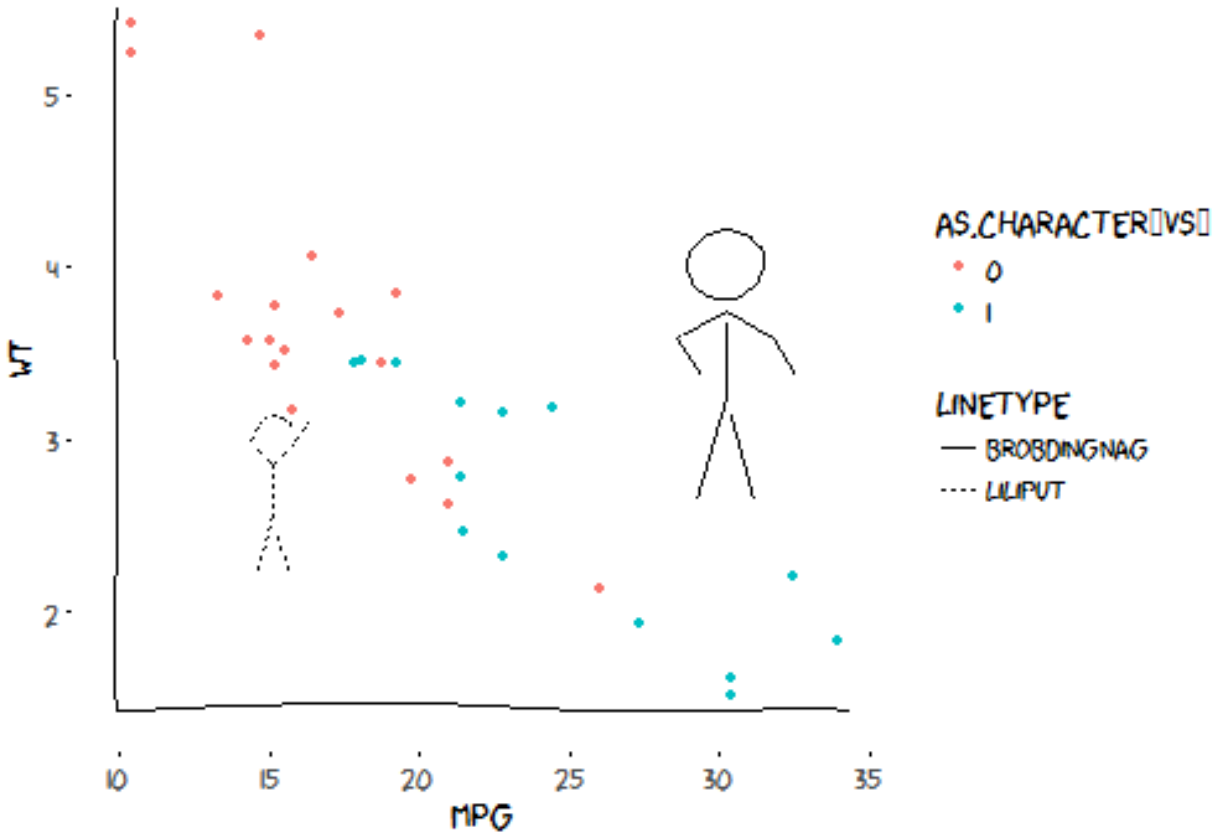

```
xrange <- range(mtcars$mpg)
yrange <- range(mtcars$wt)
p <- ggplot() + geom_point(aes(mpg, wt), data = mtcars) + xkcdaxis(xrange, yrange)
p
```



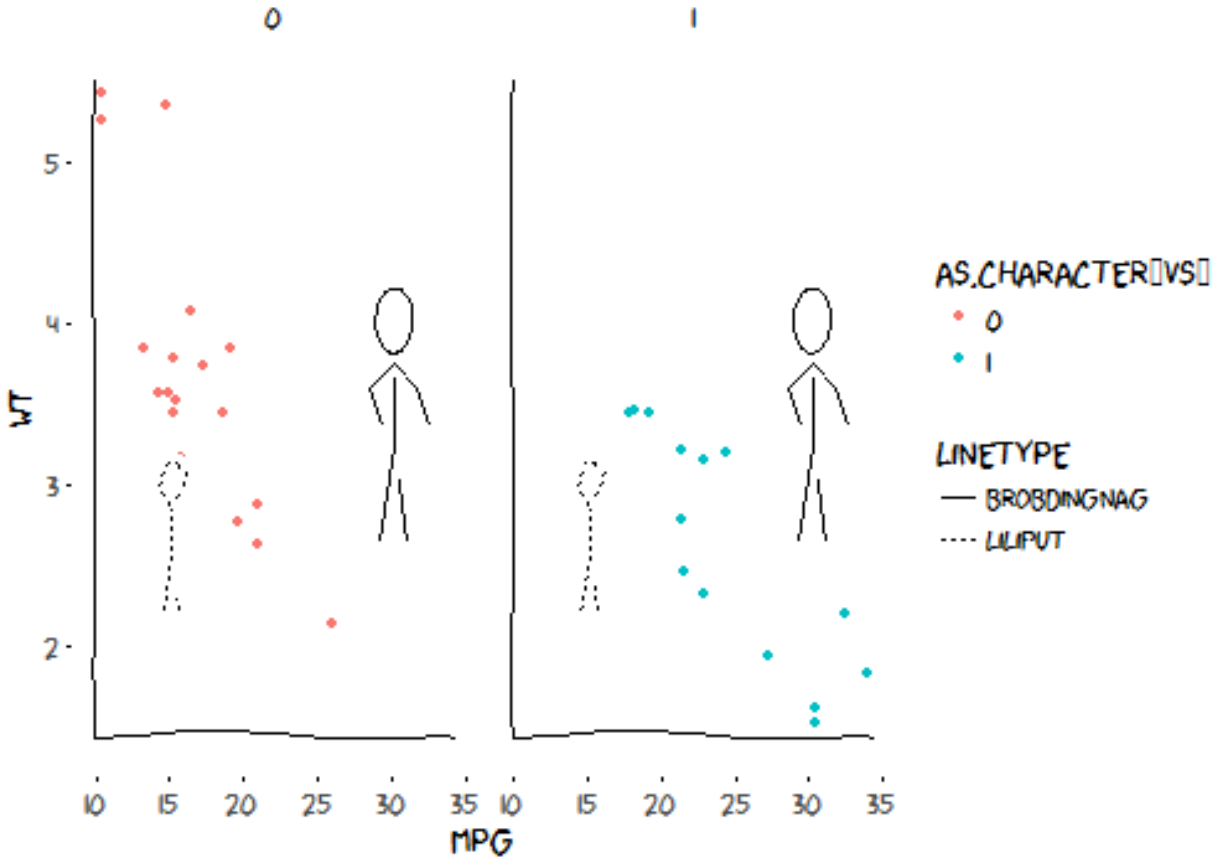

```
ratioxy <- diff(xrange)/diff(yrange)

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), 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
```




```
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), 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
```

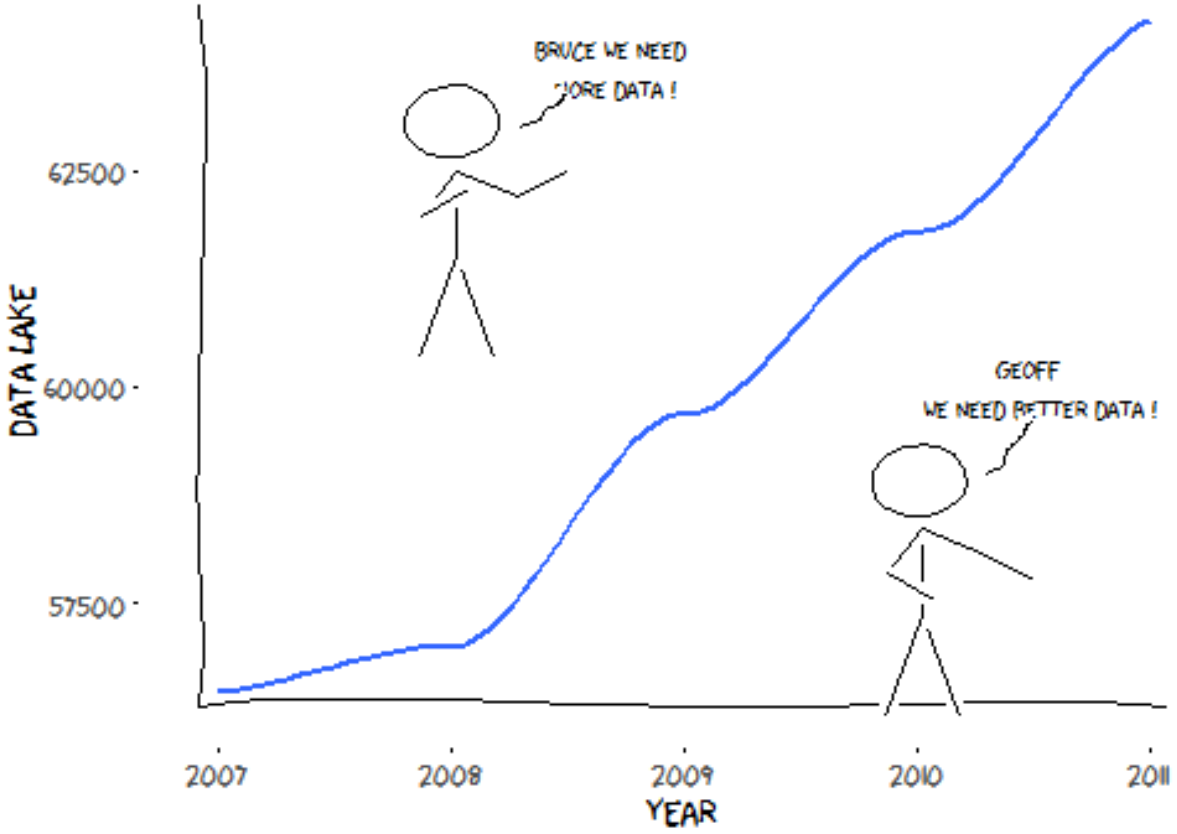



```
# Additionally, you may use the facet option of \pkg{ggplot2} to do split
# up your data by one or more variables and plot the subsets of data
# together.
p + facet_grid(. ~ vs)
```




```
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)

dataman1 <- 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("DATA LAKE ") + xkcdman(mapping,
    dataman1) + annotate("text", x = 2008.7, y = 63700, label = "Bruce We need\n more data !",
    family = "xkcd") + annotate("text", x = 2010.5, y = 60000, label = "Geoff \n  we need better data !
    family = "xkcd") + xkcdline(aes(xbegin = xbegin, ybegin = ybegin, xend = xend,
    yend = yend), datalines, xjitteramount = 0.12)
```

```
p
```




```
# ggsave() is the preferred function for saving a ggplot2 plot. For
# instance, the following ggsave('gr1.png', p)
```