# Timings of BIG data visualization with the `tabplot` package

Martijn Tennekes and Edwin de Jonge

December 16, 2013
(A later version may be available on CRAN)

**Abstract**

We test the speed of `tabplot` package with datasets over $1,00,000,000$ records. For this purpose we multiply the diamonds dataset from the `ggplot2` package 2,000 times.

# Contents

# 1 Introduction

This dataset contains 53940 records and 10 variables.

# 2 Create testdata

```
require(ggplot2)
data(diamonds)
## add some NA's
is.na(diamonds$price) <- diamonds$cut == "Ideal"
is.na(diamonds$cut) <- (runif(nrow(diamonds)) > 0.8)
```

```
n <- nrow(diamonds)
N <- 200L * n

## convert to ff format (not enough memory otherwise)
require(ffbase)
diamondsff <- as.ffdf(diamonds)
nrow(diamondsff) <- N

# fill with identical data
for (i in chunk(from = 1, to = N, by = n)) {
    diamondsff[i, ] <- diamonds
}
```

# 3 Prepare data

The preparation step is the most time consuming. Per column, the rank order is determined.

```
system.time(p <- tablePrepare(diamondsff))

##    user  system elapsed
##   43.91    9.71   95.12
```

# 4 Create tableplots

To focus on the processing time of the tableplot function, the `plot` argument is set to `FALSE`.

```
system.time(tab <- tableplot(p, maxN = 100, plot = FALSE))

##    user  system elapsed
##    5.24    2.46   13.54
```

```
system.time(tab <- tableplot(p, maxN = 1000, plot = FALSE))

##    user  system elapsed
##    5.61    1.87   11.52
```

```
system.time(tab <- tableplot(p, maxN = 10000, plot = FALSE))

##    user  system elapsed
##    5.85    1.31    7.61
```

```
system.time(tab <- tableplot(p, maxN = 1e+05, plot = FALSE))

##    user  system elapsed
##    5.96    1.26    7.65
```

```
system.time(tab <- tableplot(p, maxN = 1e+06, plot = FALSE))

##    user  system elapsed
##    6.03    1.49    7.66
```

```
system.time(tab <- tableplot(p, maxN = 0, plot = FALSE))

##    user  system elapsed
##    6.49    1.34    7.99
```