Global Drivers of Natural Forest Loss: a visual workflow for functions in the *caroline* R-package (Version 1.1.1)

David M. Schruth
`code@anthropoidea.org`

July 5, 2026

1 Introduction

Beyond use for lumber and paper products, trees provide a host of other direct and indirect benefits to human-kind. Beside from the personal benefits—bestowing us with shade and a pleasing aesthetic to our everyday visual backdrop—they can also do the more sober work of protecting fauna, preventing erosion, buffering the water-table, and performing many other ecosystem services, importantly including the production of oxygen we breathe from carbon-dioxide (normally a net-CO_2 sink in forests) [1] [2] [3] [4]. Unfortunately, the world's forests are being depleted almost entirely due to human behaviors [5]. These activities include: clearing land to feed our growing population (eg, **agriculture**), housing people by restructuring felled [woody] vegetation (eg, **lumber**), powering our increasingly technology [machine & computer] driven industries (eg, **mining**), and indirectly as an result of ([anthropogenic] global-warming fueled) **wildfires**.

Here I showcase some of the statistical (especially plotting) tools in the *caroline* (R [9]) package (version 1.1.1) [11] that serve to highlight these biodiversity-imperiling threats which have done little but increase in severity—not only throughout the Anthropocene, but notably also over the past few decades, as illustrated here by way of analyzing the Global Forest Watch dataset ([6] [7] [8]). In the following vignette, I highlight these evolving problems while expounding upon the principal findings prevised above. Further notable, while African (The Republic of Congo) and South American countries (Brazil & Columbia) have recently approached the highest of world-wide [tropical] forest loss numbers, South [East] Asian nations (Indonesia, Malaysia, Laos, Myanmar, India) and Oceania (Papua New Guinea) are rapidly over-taking them just this past decade. Furthermore, mining and (especially) wildfire are catching-up to logging and agriculture, both of which appear to be plateauing over the past few decades. These findings and more are previewed below via demonstration of the many useful functions built into the package.

My featured analysis tool, introduced here as the 'sparge plot', combines modern plotting technology features (translucent over-plotting) with the simplicity of one the oldest forms of distributional representation: the boxplot [10]. Additional plots, innovated since, have improved upon mere boxplots, including the `stripchart()` [12], the *forestplot()* [14], but these functions, like boxplots themselves, have the drawback of abstracting away too much detail. The even more rudimentary `rug()`, `stem() & leaf` plot functions attempted to show more unadulterated detail, but these remain rather limited when simultaneously representing any potentially superimposed univariate plot points. There are many plotting functions currently, however, that do a fine job of succinctly showing more distributional detail, including *sinaplot* [15], *beanplot()* [13], and *vioplot* (note that an instantiation here as `violins()` was withdrawn from *caroline* for greater perspicuity). And some packages can now even showcase both individual points and distributional smoothing summary-shapes simultaneously, as in `raincloud()` [16] and `beeswarm()` plots, although these tend to occupy more plotting space than their predecessors.

My hybrid innovation, presented in the form of the `plot.sparge()`, goes even further by revealing the myriad possible underlying nuances of distributional conformation, for even the largest of datasets, in an efficiently compact way—thanks primarily to the broad continuum of possible shades of overlapping translucent points—all while managing to circumvent engineering complexities of external smoothing, density, or positioning algorithms. The added essentialistic scaffolding of a boxplot (with auto-adjusting grayscale-shaded outlines) overlaid atop the underlying data, empowers quantitative visual undergirding in alleviation of any residual coordinate opacity, especially within the most congested areas (of the narrow rectangular strips of point-nebula) within the plot—importantly demarcating the median, quartiles, wiskers (and [thus] possible outliers) of the many point-values so routinely generated by automated systems that typify our modern, data-fertile era.

2 Software

```
> # wget https://cran.r-project.org/src/contrib/Archive/caroline/caroline_1.0.1.tar.gz
> # R CMD INSTALL caroline_1.0.1.tar.gz
```

Building the *caroline* package from source can be easily done by downloading from CRAN. Installation can typically be accomplished via the following commands from within the R command line environment:

```
install.packages('caroline')
```

After a successful installation the *caroline* package can be loaded in the normal way: by starting R and invoking the following `library` command:

```
> library(caroline)
```

No other software is required! The sparge function (dissimilar to many analogous functions: *forestplot()*, *sinaplot*, `stripchart()` , `beeswarm()`, `raincloud()`) is efficiently coded and lightweight, by depending on no other software packages (eg, tidyr, dplyr, plotly, ggplot).

3 Data

The primary dataset used here derives from the Global Forest Watch repository [6]. After minimal recoding (eg, consolidation of some loss 'driver' levels), merging, and regrouping. Below I read in the tree cover data and continents lookup list and merge them together.

```
> tl.outcome.var <- 'tc_loss_ha_log'  # the main "treeloss" outcome variable (log of total hectares lost)
> C.df <- read.csv(system.file("extdata",'countries.contients.csv', package="caroline")); C.df$X <- NULL;#, ro
> c.dup <- table(C.df$country)>1; C.df <- subset(C.df, !country%in%names(c.dup)[c.dup]); rownames(C.df)<-C.df$
> continents <- nv(C.df, 2:1)
> ## Global Forest Watch downloaded datafiles
> #       national level
> alltrees.loss.n  <- read.csv(system.file("extdata",'forest',"GFW-loss-alltrees.national-drivers.DB.csv", pac
> tropical.loss.n  <- read.csv(system.file("extdata",'forest',"GFW-loss-tropical.national-drivers.DB.csv", pac
> #   SUB-national level  (first level just be low nation: eg: state/province)
> tropical.loss.sn <- read.csv(system.file("extdata",'forest',"GFW-loss-tropical.subnatnl-drivers.DB.csv", pac
> ## three corresponding (2-column) database-style lookup tables to reduce packaged/storage size of [SubNationa
> nats.alltre.lu <- read.csv(system.file("extdata",'forest',"GFW-loss-alltrees.national-drivers-country.LU.csv
> nats.tropic.lu <- read.csv(system.file("extdata",'forest',"GFW-loss-tropical.national-drivers-country.LU.csv
> nats.trp.sn.lu <- read.csv(system.file("extdata",'forest',"GFW-loss-tropical.subnatnl-drivers-country.LU.csv
> subnats.trp.lu <- read.csv(system.file("extdata",'forest',"GFW-loss-tropical.subnatnl-drivers-subnation.LU.c
> drivers.lu     <- read.csv(system.file("extdata",'forest',"GFW-loss-driver.LU.csv", package="caroline"))  #s
> alltrees.loss.n <- nerge(all.x=T, method='lookup',## demo of the 'nerge()' and 'nv()' caroline package R fun
+   l=list(all=alltrees.loss.n,  country=nv(nats.alltre.lu),
+                                driver=nv(drivers.lu)))
> tropical.loss.n <- nerge(all.x=T, method='lookup',## demo of the 'nerge()' and 'nv()' caroline package R fun
+   l=list(tl1=tropical.loss.n,  country=nv(nats.tropic.lu),
+                                driver=nv(drivers.lu)))
> tropical.loss.sn <- nerge(all.x=T, method='lookup',## demo of the 'nerge()' and 'nv()' caroline package R fu
+   l=list(tl2=tropical.loss.sn, country=nv(nats.trp.sn.lu),
+                           subnational=nv(subnats.trp.lu),
+                                driver=nv(drivers.lu)))
>
> # an alternative piece-meal example of the above one-liner for both:#
> #  1) using nv to create vectors from lookup tables: nv(<lookuptable.lu>, name='id')
> #  2) merging the main table to these these three other lookup tables one at a time: nerge(list(df,vect))
> #tropical.loss.sn.ex <- nerge(list(tl=tropical.loss.sn   , country=    nv(nats.trp.sn.lu, 'id')), by.x='ids'
> #tropical.loss.sn.ex <- nerge(list(tl=tropical.loss.sn.ex, subnational=nv(subnats.trp.lu, 'id')), by.x='ids'
> #tropical.loss.sn.ex <- nerge(list(tl=tropical.loss.sn.ex, driver=     nv(drivers.lu, 'id')), by.x='ids', al
>
> ## examples of recoding & conversions  used in another script but in the context of the data.reconfig code b
> ## performed on the 'sub-national' dataset prior to saving as the above [GFW].csv file
> # drivers.df <- subset(drivers.df, driver!='Other natural disturbances')
> # drivers.df <- subset(drivers.df, driver!='Settlements & Infrastructure')
> # drivers.df$driver[drivers.df$driver=='Shifting cultivation'] <- 'Agriculture'
> # drivers.df$driver[drivers.df$driver=='Permanent agriculture'] <- 'Agriculture'
> # drivers.df$driver[drivers.df$driver=='Hard commodities'] <- 'Mining & Energy'
```

4 Methods

The *caroline* package has several data cleaning (`m()`), conversion `tab2df()`, and convenience functions (eg, `nv()`)—alongside several database-style merging (`nerge()`), aggregation (`groupBy()`, `bestBy()`), and reporting (`sstable()`) functions—that are routinely used throughout this vignette and the package itself. Here I use many of these to merge, reconfigure, regroup, and analyze the various tables of the Global Forest Watch data repository.

```
> ##################
> ### NATIONAL level
> ###### ALL TREES ######
> alltrees.loss.n.C <- merge(x=alltrees.loss.n, y=C.df, by='country', all.x=T) #merge tree loss data to contine
> # turn most predictors in to factors
> alltrees.loss.n.C$years     <-  cut(x=alltrees.loss.n.C$year, breaks=2002+c(0:4*6)-1)
> alltrees.loss.n.C$continent <- factor(alltrees.loss.n.C$continent)
> alltrees.loss.n.C$driver    <- factor(alltrees.loss.n.C$driver)
> all.nations.TOTs.by.C <- tapply(X=alltrees.loss.n.C[,tl.outcome.var], INDEX=list(alltrees.loss.n.C$continent,
> all.nations.TOTs.by.c <- groupBy(df=alltrees.loss.n.C, by='country',aggregation=c('min','sum'),
+                                                                        clmns=c('country',tl.outcome.var))
> alltrees.loss.n.XTonCd <- tapply(X=alltrees.loss.n.C[,tl.outcome.var],  FUN=sum,
+                                                                  INDEX=list(alltrees.loss.n.C$continent,
+                                                                             alltrees.loss.n.C$driver))
> ###### TROPICAL ######
> tropical.loss.n.C <- merge(x=tropical.loss.n, y=C.df, by='country', all.x=T) ##merge tree loss data to contin
> # turn most predictors in to factors
> tropical.loss.n.C$years     <-  cut(x=tropical.loss.n.C$year, breaks=2002+c(0:4*6)-1)
> tropical.loss.n.C$continent <- factor(tropical.loss.n.C$continent)
> tropical.loss.n.C$driver    <- factor(tropical.loss.n.C$driver)
> tropical.nations.TOTs.by.C <- tapply(X=tropical.loss.n.C[,tl.outcome.var], INDEX=list(tropical.loss.n.C$conti
> tropical.loss.n.XTonCYs <- tapply(X=tropical.loss.n.C[,tl.outcome.var],  FUN=sum,
+                                                                  INDEX=list(tropical.loss.n.C$years,
+                                                                             tropical.loss.n.C$continent))
> tropical.loss.n.XTonCd <- tapply(X=tropical.loss.n.C[,tl.outcome.var],  FUN=sum,
+                                                                  INDEX=list(tropical.loss.n.C$continent,
+                                                                             tropical.loss.n.C$driver))
> tropical.nations.TOTs.by.yr  <- groupBy(tropical.loss.n.C, by='year', aggregation=c('max','sum'), clmns=c('ye
> tropical.nations.TOTs.by.c <- groupBy(df=tropical.loss.n.C, by='country',aggregation=c('min','sum'),
+                                                                        clmns=c('country',tl.outcome.var))

> #########################
> ### SUB-NATIONAL level*
> tropical.loss.sn.C <- merge(x=tropical.loss.sn, y=C.df, by='country', all.x=T) ##merge tree loss data to con
> # turn most predictors in to factors
> tropical.loss.sn.C$years     <-  cut(x=tropical.loss.sn.C$year, breaks=2002+c(0:4*6)-1) # factorized
> tropical.loss.sn.C$continent <- factor(tropical.loss.sn.C$continent)
> tropical.loss.sn.C$driver    <- factor(tropical.loss.sn.C$driver)
> tropical.loss.sn.C$tc_loss_ha_log.f <- cut(tropical.loss.sn.C[,tl.outcome.var], breaks=c(0,3,6,9,12,15))
> tropical.subnats.TOTs.by.C <- tapply(X=tropical.loss.sn.C[,tl.outcome.var], INDEX=list(tropical.loss.sn.C$co
> tropical.subnats.XTbyc <- tab2df(table(tropical.loss.sn$country, tropical.loss.sn[,tl.outcome.var]), check.na
> tropical.subnats.XTbyc$total <- apply(tropical.subnats.XTbyc, 1, sum)
> tropical.subnats.TOTs.by.yr  <- groupBy(tropical.loss.sn.C, by='year', aggregation=c('max','sum'), clmns=c('
> tropical.subnats.TOTs.by.c  <- groupBy(tropical.loss.sn.C, by='country', aggregation=c('max','sum'), clmns=c
> tropical.subnats.TOTs.by.sn <- groupBy(tropical.loss.sn.C, by='subnational', aggregation=c('max','sum'), clm
> tropical.subnats.SST.by.yd <- sstable(x=tropical.loss.sn.C, idx.clmns=c('year','driver'), ct.clmns=tl.outcom
> tropical.subnats.SST.by.Cd <- sstable(x=tropical.loss.sn.C, idx.clmns=c('continent','driver'), ct.clmns=tl.o
> tropical.subnats.SST.by.yC <- sstable(x=tropical.loss.sn.C, idx.clmns=c('year','continent'), ct.clmns=tl.out
> ## cross-tabulate the particular sub-national regions with the most de-forestation
> tropical.loss.sn.XTonCd <- tapply(X=tropical.loss.sn.C[,tl.outcome.var],  FUN=sum,
+                                                                  INDEX=list(tropical.loss.sn.C$continent,
+                                                                             tropical.loss.sn.C$driver))
```

5 Results

Below are five different sparge plots illustrating what the *caroline* package can do to untangle confounding—using such an essentialistic implementation, less-reliant on many other packages. The first three plots use the national ("country") level data, while the much higher-resolution "sub-national" level dataset is investigated somewhat along-side for independent comparison purposes. (Note that the sparge plots have only been generated here for these smaller, national-level, datasets due to size constraints of the vector based plots in a PDF shipped in a compact CRAN package!)

```
> # printing out some summaries of the key variables of interest for each dataset
> variables.of.primary.interest <- c('years','continent','driver',tl.outcome.var)
> summary(alltrees.loss.n.C[ ,variables.of.primary.interest])

        years               continent                   driver     tc_loss_ha_log
 (2001,2007]:4502   Africa       :4843   Agriculture    :7111   Min.   : 0.000
 (2007,2013]:4568   Asia         :4220   Logging        :3947   1st Qu.: 2.830
 (2013,2019]:4304   Europe       :3827   Mining & Energy:3790   Median : 5.390
 (2019,2025]:4404   North America:2688   Wildfire       :3696   Mean   : 5.565
 NA's       : 766   Oceania      :1033                          3rd Qu.: 8.060
                    South America:1592                          Max.   :15.870
                    NA's         : 341

> summary(tropical.loss.n.C[ ,variables.of.primary.interest])

        years               continent                   driver     tc_loss_ha_log
 (2001,2007]:2128   Africa       :2952   Agriculture    :3832   Min.   : 0.000
 (2007,2013]:2152   Asia         :1999   Logging        :1676   1st Qu.: 2.080
 (2013,2019]:2154   North America:1856   Mining & Energy:1522   Median : 4.960
 (2019,2025]:2166   Oceania      : 476   Wildfire       :1570   Mean   : 4.944
                    South America:1314                          3rd Qu.: 7.410
                    NA's         :   3                          Max.   :14.460

> summary(tropical.loss.sn.C[,variables.of.primary.interest])

        years                continent                    driver      tc_loss_ha_log
 (2001,2007]:19025   Africa       :19807   Agriculture    :41761   Min.   : 0.000
 (2007,2013]:19732   Asia         :27963   Logging        :15250   1st Qu.: 1.100
 (2013,2019]:19220   North America:11359   Mining & Energy:10346   Median : 2.830
 (2019,2025]:19057   Oceania      : 3456   Wildfire       : 9677   Mean   : 3.243
                     South America:14446                           3rd Qu.: 5.070
                     NA's         :    3                           Max.   :13.750

> ## ... also peek at the continent-level percentages for each
> pct(rev(sort(     all.nations.TOTs.by.C)));

       Africa          Asia        Europe North America South America       Oceania
        "26%"         "23%"         "20%"         "13%"         "12%"          "5%"

> pct(rev(sort(tropical.nations.TOTs.by.C)));

       Africa          Asia South America North America       Oceania
        "32%"         "26%"         "22%"         "15%"          "4%"

> pct(rev(sort(tropical.subnats.TOTs.by.C)));

         Asia        Africa South America North America       Oceania
        "35%"         "26%"         "23%"         "11%"          "5%"
```

```
> fp.lines <- 50 # number of lines for printing a full page of tabular data

> ## we can also look a the nations with the most losses per dataset by their individual outcome measures
> rev(sort(all.nations.TOTs.by.C))

       Africa          Asia        Europe North America South America       Oceania
     26979.87      23514.15      20098.04      13541.55      12668.33       5327.04

> print(head(all.nations.TOTs.by.c[rev(order(all.nations.TOTs.by.c[,tl.outcome.var])),], fp.lines/4), row.name

               country tc_loss_ha_log
 Republic of the Congo        2227.67
                 India        1990.61
                Brazil        1499.93
             Indonesia        1452.68
                 China        1289.18
              Malaysia        1221.49
                Russia        1200.80
         United States        1197.37
                Canada        1169.42
               Bolivia        1158.72
                  Peru        1153.26
                México        1147.10

> rev(sort(all.nations.TOTs.by.C))

       Africa          Asia        Europe North America South America       Oceania
     26979.87      23514.15      20098.04      13541.55      12668.33       5327.04

> print(head(tropical.nations.TOTs.by.c[rev(order(tropical.nations.TOTs.by.c[,tl.outcome.var])),], fp.lines/4)

               country tc_loss_ha_log
 Republic of the Congo        1958.96
                 India        1455.86
                Brazil        1319.59
             Indonesia        1288.89
                  Peru        1073.47
              Malaysia        1072.16
               Bolivia        1044.93
             Venezuela         914.82
      Papua New Guinea         914.59
                México         903.06
               Myanmar         899.64
              Colombia         893.82

> rev(sort(all.nations.TOTs.by.C))

       Africa          Asia        Europe North America South America       Oceania
     26979.87      23514.15      20098.04      13541.55      12668.33       5327.04

> print(head(tropical.subnats.TOTs.by.c[rev(order(tropical.subnats.TOTs.by.c[,tl.outcome.var])),], fp.lines/4)

               country tc_loss_ha_log
             Indonesia       20311.82
 Republic of the Congo       18362.78
                Brazil       12582.17
               Vietnam       12223.80
           Philippines       10221.51
              Colombia        9317.12
      Papua New Guinea        8807.41
                  Laos        7837.11
                  Peru        7499.82
              Thailand        7472.32
                México        6715.77
              Malaysia        6389.88
```

```
> ## Let's merge together all of the national totals for losses using the three different datasets'
> all3dfs <- list(a.n=all.nations.TOTs.by.c, t.n=tropical.nations.TOTs.by.c, t.sn=tropical.subnats.TOTs.by.c)
> ## we want to first just extract a single 'loss,ha' column and name it by country
> all3dfs.vl <- lapply(all3dfs, nv, name=c(tl.outcome.var, 'country'))
> ## we can then "nerge()" them together using my so-named name-merge function
> TOTs.by.c <- nerge(all3dfs.vl)#, method='rownames')
> ## we can now use the 'pct()' function to create column wise percentages to better compare across columns
> PTCs.by.c <- pct(TOTs.by.c, clmns=names(TOTs.by.c))
> PTCs.by.c <- PTCs.by.c[,grepl('pct',names(PTCs.by.c))] *100
> PTCs.by.c$avg <- apply(PTCs.by.c[,1:2], 1, mean) # avg of just the first two 'national' level columns
> ## merge the finished percentage table back with the continents
> PTCs.by.cC<- nerge(list(pcts=round(PTCs.by.c,1), continents=continents)) #conts=C.df))
> head(PTCs.by.cC[rev(order(PTCs.by.cC$avg)),], fp.lines*3/4)
```

	a.n.pct	t.n.pct	t.sn.pct	avg	continents
Republic of the Congo	3.4	4.6	7.4	4.0	Africa
India	3.0	3.4	2.3	3.2	Asia
Brazil	2.3	3.1	5.0	2.7	South America
Indonesia	2.2	3.0	8.1	2.6	Asia
Malaysia	1.9	2.5	2.6	2.2	Asia
Peru	1.8	2.5	3.0	2.1	South America
Bolivia	1.8	2.5	1.8	2.1	South America
Venezuela	1.7	2.2	2.1	1.9	South America
Papua New Guinea	1.6	2.2	3.5	1.9	Oceania
México	1.8	2.1	2.7	1.9	North America
Myanmar	1.7	2.1	2.3	1.9	Asia
Colombia	1.6	2.1	3.7	1.9	South America
Vietnam	1.7	2.0	4.9	1.8	Asia
Madagascar	1.7	2.0	1.3	1.8	Africa
Laos	1.6	2.0	3.1	1.8	Asia
China	2.0	1.6	0.6	1.8	Asia
Cameroon	1.5	2.1	1.6	1.8	Africa
Guyana	1.4	1.9	1.7	1.7	South America
Cambodia	1.5	1.9	2.5	1.7	Asia
Suriname	1.3	1.8	1.1	1.6	South America
Philippines	1.5	1.8	4.1	1.6	Asia
Guatemala	1.5	1.8	1.4	1.6	North America
Gabon	1.3	1.8	1.7	1.6	Africa
Argentina	1.6	1.7	1.0	1.6	South America
Angola	1.6	1.5	1.2	1.6	Africa
Paraguay	1.4	1.6	1.7	1.5	South America
Liberia	1.3	1.7	2.1	1.5	Africa
Ecuador	1.4	1.7	1.6	1.5	South America
Central African Republic	1.4	1.7	1.7	1.5	Africa
Thailand	1.4	1.4	3.0	1.4	Asia
Tanzania	1.6	1.2	0.9	1.4	Africa
Nigeria	1.2	1.5	1.1	1.4	Africa
Ghana	1.3	1.5	1.0	1.4	Africa
Nicaragua	1.2	1.4	0.8	1.3	North America
Honduras	1.3	1.3	1.3	1.3	North America
Belize	1.1	1.5	0.8	1.3	North America
Zambia	1.6	0.8	0.2	1.2	Africa

```
> ## Above is a table of column-wise percentages of loss ordered by the "avg" column,
> ##  which is merely an average of the first two *national*level* columns. The third
> ##  *sub-national percentage column is merely here for informational/comparison reasons.
```

```
> # now just look at the top best(inverse=T)->[worst] 5 countries in each continent
> PTCs.by.cC$country <- rownames(PTCs.by.cC)
> BB.PTC.cC <- bestBy(df=PTCs.by.cC, by='continents', best='avg', top=6, rebind=F, inverse=T)
> print(BB.PTC.cC, row.names=F)

$Africa
 a.n.pct t.n.pct t.sn.pct avg continents                  country
     3.4     4.6      7.4 4.0     Africa    Republic of the Congo
     1.5     2.1      1.6 1.8     Africa                 Cameroon
     1.7     2.0      1.3 1.8     Africa               Madagascar
     1.6     1.5      1.2 1.6     Africa                   Angola
     1.3     1.8      1.7 1.6     Africa                    Gabon
     1.4     1.7      1.7 1.5     Africa Central African Republic

$Asia
 a.n.pct t.n.pct t.sn.pct avg continents   country
     3.0     3.4      2.3 3.2       Asia     India
     2.2     3.0      8.1 2.6       Asia Indonesia
     1.9     2.5      2.6 2.2       Asia  Malaysia
     1.7     2.1      2.3 1.9       Asia   Myanmar
     2.0     1.6      0.6 1.8       Asia     China
     1.6     2.0      3.1 1.8       Asia      Laos

$`North America`
 a.n.pct t.n.pct t.sn.pct avg    continents   country
     1.8     2.1      2.7 1.9 North America    México
     1.5     1.8      1.4 1.6 North America Guatemala
     1.1     1.5      0.8 1.3 North America    Belize
     1.3     1.3      1.3 1.3 North America  Honduras
     1.2     1.4      0.8 1.3 North America Nicaragua
     1.1     1.4      1.1 1.2 North America    Panama

$Oceania
 a.n.pct t.n.pct t.sn.pct avg continents          country
     1.6     2.2      3.5 1.9    Oceania Papua New Guinea
     1.0     1.2      0.8 1.1    Oceania  Solomon Islands
     1.6     0.0      0.0 0.8    Oceania        Australia
     0.6     0.4      0.1 0.5    Oceania             Fiji
     0.5     0.4      0.1 0.4    Oceania          Vanuatu
     0.2     0.1      0.0 0.1    Oceania            Palau

$`South America`
 a.n.pct t.n.pct t.sn.pct avg    continents   country
     2.3     3.1      5.0 2.7 South America    Brazil
     1.8     2.5      1.8 2.1 South America   Bolivia
     1.8     2.5      3.0 2.1 South America      Peru
     1.6     2.1      3.7 1.9 South America  Colombia
     1.7     2.2      2.1 1.9 South America Venezuela
     1.4     1.9      1.7 1.7 South America    Guyana

> ## Above is a version of of the forest loss percentage table that has been broken down by
> ##  each continent, ranked within each sub section (using bestBy()) to highlight the top
> ##  six highest forest losses for each country within in each continent separately.
```

```
> ## Now let's look at how forest loss happens (for TROPICAL NATIONS ONLY) over time'
> # we've already created a cross-tabulation of loss by continent over the years
> tropical.loss.n.XTonCYs

             Africa    Asia North America Oceania South America
(2001,2007] 2760.50 2674.00       1467.73  398.18       2216.15
(2007,2013] 3066.71 2840.22       1563.66  457.48       2335.12
(2013,2019] 3722.59 2855.01       1712.22  509.87       2441.57
(2019,2025] 3860.39 2882.15       1726.85  452.52       2572.16

> # let's modify this to a finer level of partitioning so that the dirvers are included split by continents
> tropical.loss.n.XTonYsdC <- tapply(X=tropical.loss.n.C[,tl.outcome.var],  FUN=sum,
+                                                                         INDEX=list(tropical.loss.n.C$years,
+                                                                                    tropical.loss.n.C$driver,
+                                                                                    tropical.loss.n.C$continent))
> tropical.loss.n.XTonYsdC

, , Africa

            Agriculture Logging Mining & Energy Wildfire
(2001,2007]     1711.84  556.72          249.98   241.96
(2007,2013]     1841.99  603.00          336.80   284.92
(2013,2019]     2082.53  738.81          463.76   437.49
(2019,2025]     2074.81  691.08          551.12   543.38

, , Asia

            Agriculture Logging Mining & Energy Wildfire
(2001,2007]     1373.83  616.65          376.58   306.94
(2007,2013]     1409.35  643.30          460.11   327.46
(2013,2019]     1375.26  633.91          489.58   356.26
(2019,2025]     1363.36  623.39          523.71   371.69

, , North America

            Agriculture Logging Mining & Energy Wildfire
(2001,2007]      954.38  160.38          111.45   241.52
(2007,2013]      998.77  187.61          128.85   248.43
(2013,2019]     1029.80  185.50          160.12   336.80
(2019,2025]      990.01  193.11          171.30   372.43

, , Oceania

            Agriculture Logging Mining & Energy Wildfire
(2001,2007]      177.87  150.44           31.34    38.53
(2007,2013]      210.38  158.16           39.17    49.77
(2013,2019]      217.39  166.58           46.49    79.41
(2019,2025]      192.08  154.89           52.24    53.31

, , South America

            Agriculture Logging Mining & Energy Wildfire
(2001,2007]     1070.70  417.97          362.03   365.45
(2007,2013]     1096.41  436.53          399.29   402.89
(2013,2019]     1104.49  463.70          439.02   434.36
(2019,2025]     1119.21  480.33          465.52   507.10

> ## the loss driver by years grouped by continent (tabular) format above can also be realized as
> ##  a sparge plot showing all underlying datapoints (see the last plot [Fig.3] below for an example)
```

```
> ## plotting results defaults
> axis.limit.outcome.an <- c(1, max(alltrees.loss.n[,tl.outcome.var])) # log(treeloss, ha)
> axis.limit.outcome.tn <- c(1, max(tropical.loss.n[,tl.outcome.var])) # log(treeloss, ha)
> axis.limit.outcome.sn <- c(1, max(tropical.loss.sn[,tl.outcome.var]))# log(treeloss, ha)
> ## last minue outcome and plot setup
> driver.cols <- nv(c('green','brown','purple','red'), levels(tropical.loss.sn.C$driver))
> # main titles for the heat matrix plots
> main.forest <- nv(c('All Trees (Worldwide)','Humid Tropical Primary Forest'), c('all', 'trp'))
> main.level <- nv(c("(@ national level)",'(@ sub-national level)'), c('ntnl','subn'))
> main.unit <- "loss in log(ha)"

> ###  HEATMATRIX PLOTS FOR ALL THREE DATASETS: [SUB-]NATIONAL & ALL/TROPICAL
> par(mfrow=c(3,1), mar=c(3,6,3,1), mgp=c(2.5,2,.5) )
> heatmatrix(round(alltrees.loss.n.XTonCd ), text.col='chartreuse', cex=1.1,
+     main=paste(main.forest['all'], "\n", main.unit, main.level['ntnl']))
> heatmatrix(round(tropical.loss.n.XTonCd ), text.col='chartreuse', cex=1.1,
+     main=paste(main.forest['trp'], "\n", main.unit, main.level['ntnl']))
> heatmatrix(round(tropical.loss.sn.XTonCd), text.col='chartreuse', cex=1.1,
+     main=paste(main.forest['trp'], "\n", main.unit, main.level['subn']))

> ###  SPARGE PLOTS FOR THE *NATIONAL* LEVEL
> ## NATIONAL LEVEL DATASET ##
> # sparge plot: forest loss (@ national level): vs continent grouped by loss driver
> par(las=1, cex=.7, mar=c(3,6,1,1), mgp=c(1.6,.7,0))
> model.4 <- paste(tl.outcome.var, 'continent | driver', sep='~')
> pds.4 <- plot.sparge(x=tropical.loss.n.C, f=model.4, xlim=axis.limit.outcome.tn,
+             xlab='log(forest loss, ha)', ylab='', pt.cols=driver.cols, main='tropical nations',
+             legend.cex=.6, legend.inset=.008, legend.title='loss driver',
+             boxplot.notch=TRUE, boxplot.lwd=1.5, boxplot.col=rgb(0,0,1,.2))
> lines.sparge(x=tropical.loss.n.C, f=model.4, pds=pds.4, cat.order=2:1, lwd=1.5, pt.cex=2,col=gray(.5))

> # sparge plot: forest loss (@ national level): vs years grouped by loss driver
> par(las=1, cex=.6, mar=c(3,3,1,1), mgp=c(1.6,.7,0)); horiz.5 <- FALSE
> model.5 <- paste(tl.outcome.var, 'years | driver', sep='~');
> pds.5 <- plot.sparge(x=tropical.loss.n.C, f=model.5, horiz=horiz.5, ylim=axis.limit.outcome.tn,
+                   pt.cols=driver.cols, xlab='years', ylab='log(forest loss, ha)', las=1,
+                      legend.cex=.7 , legend.inset=.112, legend.title='loss driver', main='tropical nations'
+                         boxplot.notch=TRUE, boxplot.lwd=1.5, boxplot.col=rgb(0,0,1,.2))
> ## add some per-group median trendline overlays (using the returned p[oosition]d[odge]s above)
> lines.sparge(x=tropical.loss.n.C, f=model.5, pds=pds.5, rb=driver.cols, lty=1, lwd=.8, horiz=horiz.5)
> lines.sparge(x=tropical.loss.n.C, f=model.5, pds=pds.5, cat.order=2:1, horiz=horiz.5, lwd=1.7, col=gray(.5))
```

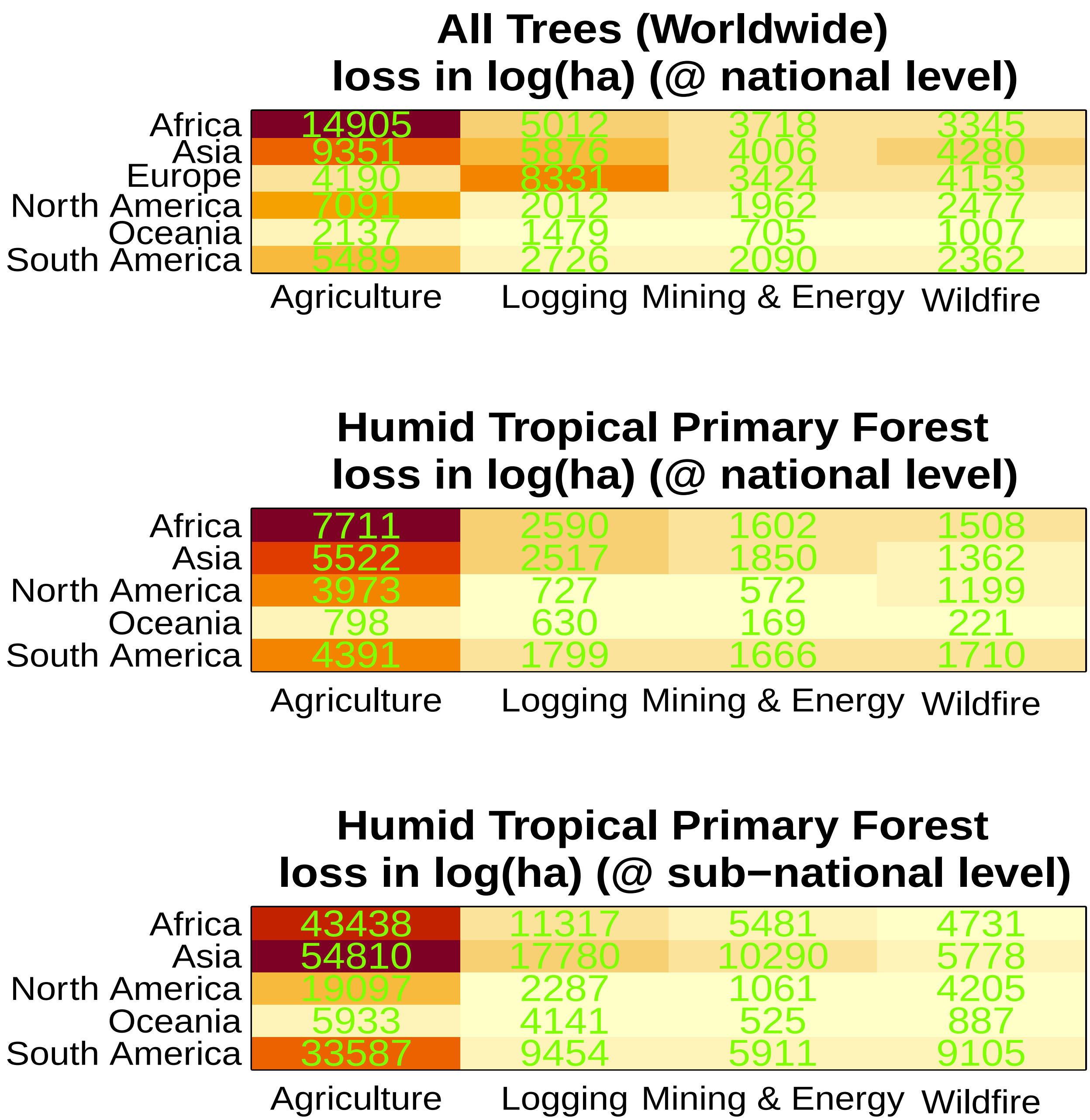

Figure 1: Three 'heatmatrix()' plots showing the (intersection of continent by drivers of) types of tree cover loss for each continent. Note that Agriculture dominates all other forms, with African (and possibly also Asian*) agriculture typically ranking highest among world-wide forest loss. The continent of Europe (eg, in Finland and Sweden) non-tropical forestry also registers as a not insignifcant manifestation of planetary-scale losses.

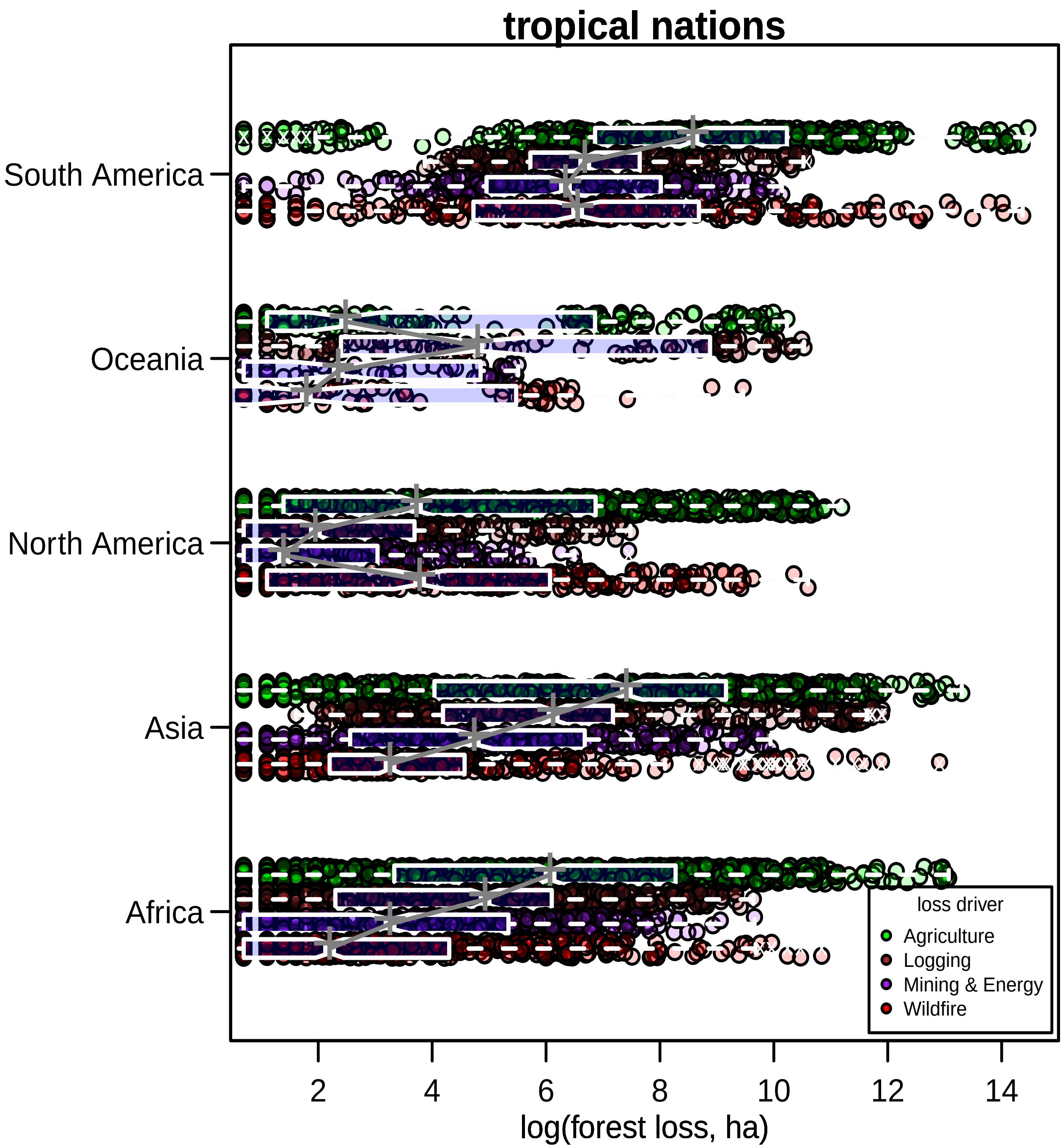

Figure 2: A sparge plot of forest loss (at the national level): vs continent grouped by loss driver. Despite the relatively low loss numbers for North America, do observe the unusual trending of logging in Oceania whose inter-quartile range is apparently outstripping all forms of loss in N. America. (Note that these point represent [˜200x24=] thousands of individual nation-year records)

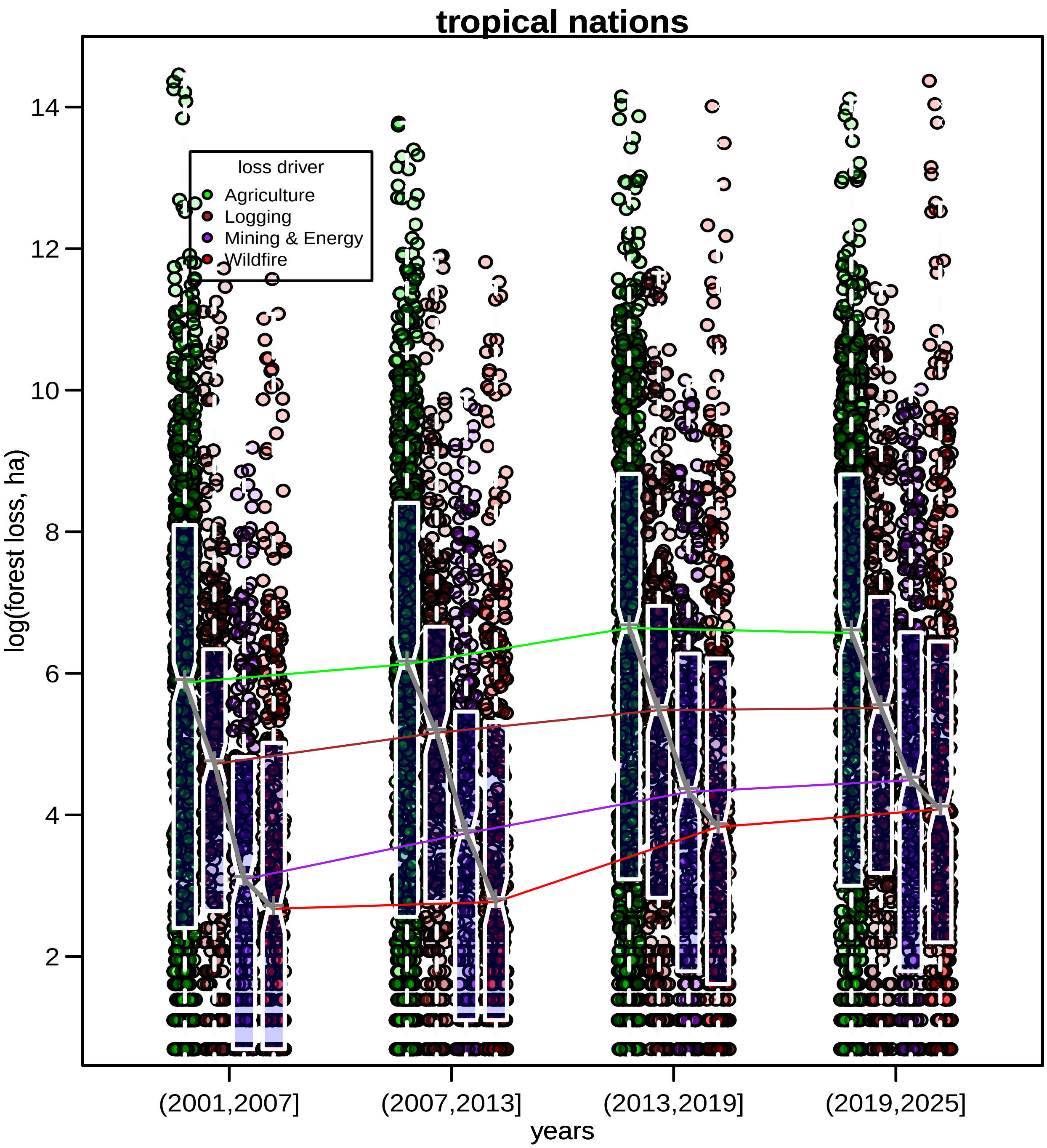

Figure 3: A [vertical] sparge plot of forest loss (at the national level): vs years grouped by loss driver. The medians ("+") of each box plot for each subgroup have been externally joined together via dashed lines showing increasing trends for our four primary drivers of tree loss. (Note that these points represent [~200x24=] thousands of individual nation-year records)

6 Conclusions

In addition to showcasing several of the *caroline* package functions (eg, `m()`, `nv()`, `pct()`, `tab2df()`, `nerge()`, `groupBy()`, `bestBy()`, `sstable()`, `heatmatrix()`), I have also introduced the `sparge` plot anew [17], as a possible apotheosis of visual-analytics which promises to facilitate the untangling of deeply confounding data. Using relatively simple formula-based calls to `plot.sparge()`, in conjunction with appropriate merges to (eg, continent) lookup table(s) (via `nerge()`), I have shown the capabilities of this statistical software package to import, clean, re-configure, analyze, and plot larger datasets.

En route, I have re-confirmed what other forestry researchers have observed: that while the dual quandary of the Congo and Amazon rain-forests are of predominate environmental concern [18], many other nations—including developing tropical countries (India, Mexico, & Panama) as well as more temperate behemoths (China, Russia, & U.S.)—also rank among the world's worst in hosting forest losses. This research has also discovered that—the world's (tropical) trees may have been most heavily depleted, over the past several decades, not only in the countries of Africa (Republic of Congo & Madagascar), South America (esp. Brazil, Columbia, Peru, Venezuela, & Bolivia), as well as more populous countries in Asia (China & India), but also many many smaller ones in South East Asia (incl. Malaysia, Laos, Vietnam, Philippines, Thailand, Cambodia, & Myanmar). And a possibly even more distressing form of distributed depletion is increasingly appearing in more remote island-based countries (eg, of Oceania) that presumably have been previously prohibitively difficult to explore, develop, survey, appraise, harvest/mine, and ship-from internationally–most especially Indonesia and Papua New Guinea.

Furthermore, while the primary drivers of world-wide losses have shifted from agriculture and logging (which both seem to be plateauing) to wildfires and minerals & energy (both still increasing) over this past decade, logging in Oceania appears to be out-pacing all other forms of loss (in Oceania) and, for comparison, even beyond all drivers in [tropical] North America. And, less surprisingly, forest loss due to [rice] agriculture in [south-east] Asia only remains overshadowed by the persistently undeniable environmental-discordance inherent to much of South American [ranching] agriculture—both often deploying slash-and-burn methods to clear land of native trees. Hopefully with increased (internet-based) access to larger higher-resolution datasets (such as the Global Forest Watch data repository, utilized extensively in this report) along with further democratization of computational access and globally-open communication, such remote reporting of these trends, can be reflected back to citizens of these countries so as to raise a counter-balanced awareness toward the potentially predestinate concomitants in such loss—that of persistent climate alteration and biome degradation.

7 Licensing

The *caroline* package is licensed under the Artistic License v2.0: it is therefore free to use and redistribute, however, we, the copyright holders, wish to maintain primary control over any further development. Please be sure to cite *caroline* if you use the package in presentations or work leading to publication.

References

[1] The Arbor Day Foundation https://www.arborday.org/value

[2] The Environmental Protection Agency https://www.epa.gov/

[3] The Nature Conservancy https://www.nature.org/en-us/

[4] Harris, Nancy, and David Gibbs. "Forests absorb twice as much carbon as they emit each year." World Resources Institute 21 (2021). https://www.wri.org/

[5] Eckholm, E. (1979). Planting for the Future: Forestry for Human Needs *Worldwatch Paper*, 26.

[6] Global Forest Watch (2026).*Global forest watch*. World Resources Institute, Washington, DC https://www.globalforestwatch.org.

[7] Hansen, M.C., P.V. Potapov, R. Moore, et al. (2013) "High-Resolution Global Maps of 21st-Century Forest Cover Change." *Science* 342: 850–53. Data available on-line from: https://glad.earthengine.app/view/global-forest-change.

[8] Sims, M., R. Stanimirova, A. Raichuk, M. Neumann et al. (2025) "Global Drivers of Forest Loss at 1 km Resolution." *Environmental Research Letters* 20 (7): 074027. DOI: https://doi.org/10.1088/1748-9326/add606.

[9] R Core Team (2026) 'R: A language and environment for statistical computing' *R Foundation for Statistical Computing*, Vienna, Austria https://www.R-project.org.

[10] McGill, R., Tukey, J. W., & Larsen, W. A. (1978). 'Variations of box plots.' *The american statistician*, 32(1), 12-16.

[11] Schruth, D.M. (2026) 'caroline: A Collection of Database, Data Structure, Visualization, and Utility Functions for R' *R package*, version 1.1.0 https://CRAN.R-project.org/package=caroline.

[12] Petreshock, J. G., Wolk, S. J., & Cresitello-Dittmar, M. (2004). *The Data Inspector: Data Visualization for the Chandra Monitoring Project*. In Astronomical Data Analysis Software and Systems IX (Vol. 216, p. 475).

[13] Kampstra, P. (2008). *Beanplot: A boxplot alternative for visual comparison of distributions*. Journal of Statistical Software, 28, 1–9. https://CRAN.R-project.org/package=beanplot

[14] Boyles, A. L., Harris, S. F., Rooney, A. A., & Thayer, K. A. (2011). *Forest Plot Viewer: a new graphing tool*. Epidemiology, 22(5), 746-747. https://CRAN.R-project.org/package=forestplot

[15] Sidiropoulos, N., Sohi, S. H., Pedersen, T. L., Porse, B. T., Winther, O., Rapin, N., & Bagger, F. O. (2018) 'SinaPlot: an enhanced chart for simple and truthful representation of single observations over multiple classes. ' *Journal of Computational and Graphical Statistics*, 27(3), 673-676. https://CRAN.R-project.org/package=sinaplot

[16] Allen, M., Poggiali, D., Whitaker, K., Marshall, T. R., Van Langen, J., & Kievit, R. A. (2021) 'Raincloud plots: a multi-platform tool for robust data visualization' *Wellcome open research*, 4, 63. https://github.com/RainCloudPlots/RainCloudPlots.

[17] Schruth, David M., Christopher N. Templeton, Darryl J. Holman, Eric A. Smith, (2024) "The origins of musicality in the motion of primates", American Journal of Biological Anthropology 184 (1) e24891 DOI: https://doi.org/10.1002/ajpa.24891.

[18] Turubanova, S., Potapov, P.V., Tyukavina, A. and Hansen, M.C., (2018) Ongoing primary forest loss in Brazil, Democratic Republic of the Congo, and Indonesia. *Environmental Research Letters*, 13(7), p.074028.