

ausplotsR: quickstart guide to basic analysis of TERN AusPlots vegetation data

Greg Guerin & Bernardo Blanco-Martin

2020-09-23

Introduction

TERN AusPlots is a national plot-based terrestrial ecosystem surveillance monitoring method and dataset for Australia (Sparrow et al. 2020). Through **ausplotsR**, users can directly access AusPlots data collected by on-ground observers on vegetation and soils, including physical sample/voucher details and barcode numbers. The dataset can be downloaded in its entirety or as individual modules, and can be subsetting by geographic bounding box or species name search. The package also includes a series of bespoke functions for working with AusPlots data, including visualisation, creating tables of species composition, and calculation of tree basal area, fractional cover or vegetation cover by growth form/structure/strata and so on.

This is a short guide for getting started with analysis of AusPlots data through the **ausplotsR** R package. More information on making use of AusPlots data in **ausplotsR** is available through the package help files and manual. Below, we demonstrate installing the package, accessing some AusPlots data, generating matrices and running simple example analyses.

More comprehensive tutorials on accessing and analysing AusPlots data (Blanco-Martin 2019) are available at: https://github.com/ternaustralia/TERN-Data-Skills/tree/master/EcosystemSurveillance_PlotData

Installing the package and accessing raw data

The latest version of **ausplotsR** can be installed directly from github using the **devtools** package, which must be installed first.

```
library(devtools)
install_github("ternaustralia/ausplotsR", build_vignettes = TRUE, dependencies = TRUE)
```

Once installed, load the package as follows. Note, packages **vegan**, **maps** and **mapdata** are required for **ausplotsR** to load, and functions are also imported from packages: **plyr**, **R.utils**, **simba**, **httr**, **jsonlite**, **sp**, **maptools**, **ggplot2**, **gtools**, **jose**, **curl** and **betapart**, while **knitr**, **rmarkdown** and **goeveg** are required to build this package vignette (i.e., if 'build_vignettes' is set to TRUE above).

```
library(ausplotsR)
```

We can now access live data, starting here with basic site information and vegetation point-intercept modules and using a bounding box to spatially filter the dataset to central Australia. All data modules are extracted via a single function, **get_ausplots**:

```
# See ?get_ausplots to explore all data modules available

my.ausplots.data <- get_ausplots(bounding_box = c(125, 140, -40, -10))
```

The output of the above call is a list with the following **\$elements**:

```
names(my.ausplots.data)
#> [1] "site.info" "veg.vouch" "veg.PI" "citation"
```

The 'site.info' table contains basic site and visit details. Here are a selected few of the many fields:

```
head(my.ausplots.data$site.info[, c("site_location_name", "site_unique", "longitude",
  "latitude", "bioregion_name")])
#>   site_location_name      site_unique longitude  latitude bioregion_name
#> 1      NTADAC0001 NTADAC0001-53518  130.7779 -13.15835      DAC
#> 2      NTASSD0015 NTASSD0015-53565  135.6168 -25.12393      SSD
#> 3      QDAMII0002 QDAMII0002-53546  138.1606 -20.00789      MII
#> 4      SATSTP0005 SATSTP0005-53513  138.8488 -29.45660      STP
#> 5      SATSTP0005 SATSTP0005-58639  138.8488 -29.45660      STP
#> 6      NTTDAB0001 NTTDAB0001-53580  131.6740 -13.96288      DAB
```

Each survey is identified by the 'site_unique' field, which is unique combination of site ID ('site_location_name') and visit ID ('site_location_visit_id'). The 'site_unique' field therefore links all tables returned from the `get_ausplots` function.

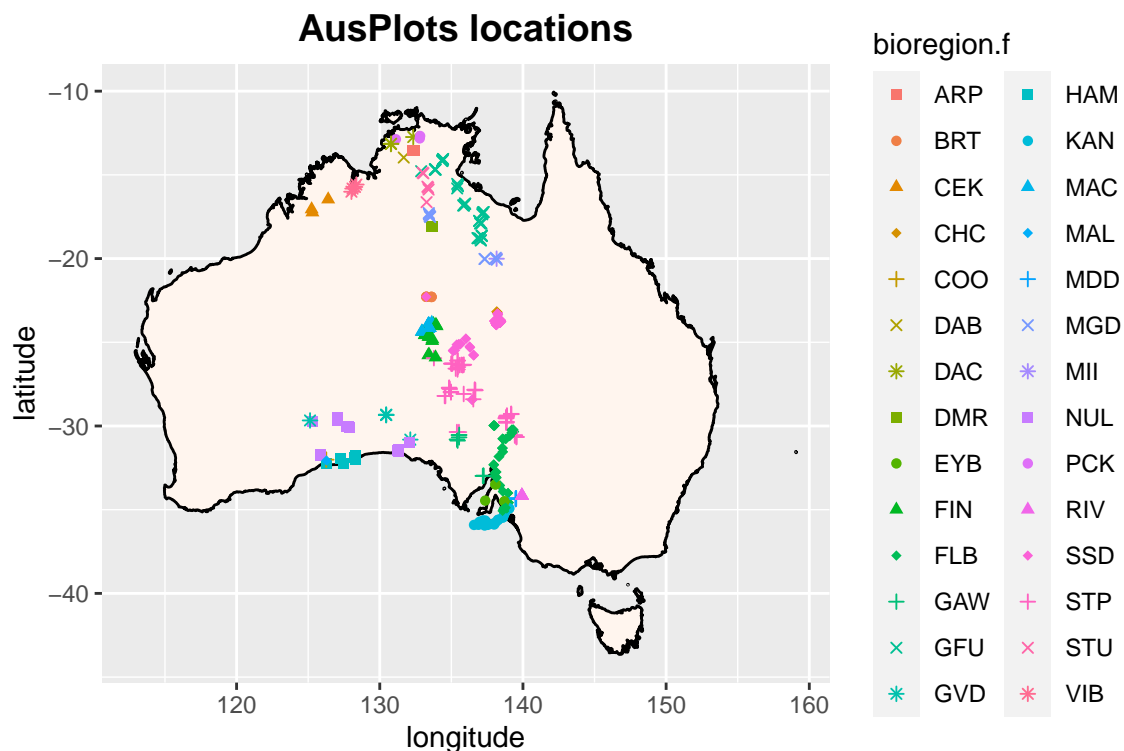
The 'site.info' table can be used to identify, subset or group surveys in space and time, for example:

```
# count plot visits per Australian States:
summary(as.factor(my.ausplots.data$site.info$state))
#> NT QLD SA WA
#> 151 48 172 25
```

Map AusPlots sites and visualise data

The package has an in-built function - see `?ausplots_visual` - to rapidly map AusPlots over Australia and to visualise the relative cover/abundance of green vegetation, plant growth forms and species. Maps can also be generated manually using the longitude and latitude fields in the `$site.info` table.

```
# Sites are coded by IBRA bioregion by default.
map_ausplots(my.ausplots.data)
```



Alternatively, the following call generates a pdf with a map of all sites and attribute graphics for selected AusPlots: `ausplotsR::ausplots_visual()`

Here is a snippet of the raw point-intercept data that will be used in the following examples to derive vegetation attributes:

```
head(subset(my.ausplots.data$veg.PI, !is.na(herbarium_determination)))
#>   site_location_name site_location_visit_id transect point_number
#> 312      NTAMGD0002           53466      E5-W5           46
#> 315      NTAMGD0002           53466      E5-W5           50
#> 339      NTAMGD0002           53466      E5-W5           87
#> 351      NTAMGD0002           53466      W4-E4            5
#> 361      NTAMGD0002           53466      W4-E4           26
#> 378      NTAMGD0002           53466      W4-E4           50
#>   herbarium_determination substrate in_canopy_sky dead growth_form height
#> 312      Cleome viscosa      Gravel      FALSE FALSE      Forb 0.10
#> 315      Cleome viscosa      Bare      FALSE FALSE      Forb 0.05
#> 339      Cleome viscosa      Bare      FALSE FALSE      Forb 0.20
#> 351      Cleome viscosa      Bare      FALSE FALSE      Forb 0.20
#> 361      Cleome viscosa      Bare      FALSE FALSE      Forb 0.10
#> 378      Portulaca digyna      Gravel      FALSE FALSE      Forb 0.02
#>   veg_barcode standardised_name      family      genus specific_epithet
#> 312 NTA 006351      Cleome viscosa      Cleomaceae      Cleome      viscosa
#> 315 NTA 006351      Cleome viscosa      Cleomaceae      Cleome      viscosa
#> 339 NTA 006351      Cleome viscosa      Cleomaceae      Cleome      viscosa
#> 351 NTA 006351      Cleome viscosa      Cleomaceae      Cleome      viscosa
#> 361 NTA 006351      Cleome viscosa      Cleomaceae      Cleome      viscosa
#> 378 NTA 006311      Portulaca digyna      Portulacaceae      Portulaca      digyna
#>   infraspecific_rank infraspecific_epithet taxa_status      genus_species authorship
#> 312      <NA>      <NA>      Accepted      Cleome viscosa      L.
#> 315      <NA>      <NA>      Accepted      Cleome viscosa      L.
```

```

#> 339          <NA>          <NA> Accepted  Cleome viscosa      L.
#> 351          <NA>          <NA> Accepted  Cleome viscosa      L.
#> 361          <NA>          <NA> Accepted  Cleome viscosa      L.
#> 378          <NA>          <NA> Unchecked Portulaca digyna  F.Muell.
#>          published_in  rank hits_unique  site_unique
#> 312  Sp. Pl. 672 1753 SPECIES  E5-W5 46 NTAMGD0002-53466
#> 315  Sp. Pl. 672 1753 SPECIES  E5-W5 50 NTAMGD0002-53466
#> 339  Sp. Pl. 672 1753 SPECIES  E5-W5 87 NTAMGD0002-53466
#> 351  Sp. Pl. 672 1753 SPECIES  W4-E4 5 NTAMGD0002-53466
#> 361  Sp. Pl. 672 1753 SPECIES  W4-E4 26 NTAMGD0002-53466
#> 378 Fragm. 1: 170 1859 SPECIES  W4-E4 50 NTAMGD0002-53466

```

Note that ‘veg_barcode’ links species hits to the vegetation vouchers module, while the ‘hits_unique’ field identifies the individual point-intercept by transect and point number (see `help(ausplotsR)` and references for more details on the plot layout and survey method). At each point, plant species (if any), growth form and height are recorded along with substrate type.

Example 1: latitudinal pattern in proportional vegetation cover

Let’s visualise basic vegetation cover as a function of latitude. First, we call the `fractional_cover` function on the extracted point-intercept data (`$veg.PI`). The function converts the raw data to proportional cover of green/brown vegetation and bare substrate. Note the calculation may take a few minutes for many AusPlots, so for this example we will pull out a subset of 100 randomly drawn sites to work with.

```

sites100 <- my.ausplots.data$veg.PI[which(my.ausplots.data$veg.PI$site_unique %in%
  sample(my.ausplots.data$site.info$site_unique, 100)), ]
my.fractional <- fractional_cover(sites100)

```

```

head(my.fractional)
#>          site_unique  bare brown green NA.
#> NTAARP0003-58424  NTAARP0003-58424 25.45 40.50 34.06 0
#> NTABRT0001-53616  NTABRT0001-53616 18.22 33.27 48.51 0
#> NTABRT0003-53618  NTABRT0003-53618 14.65 30.59 54.75 0
#> NTABRT0004-53619  NTABRT0004-53619 22.10 31.42 46.48 0
#> NTABRT0005-53620  NTABRT0005-53620 36.63 19.41 43.96 0
#> NTAFIN0002-53622  NTAFIN0002-53622  5.63 19.15 75.22 0

```

Next, we need to merge the fractional cover scores with longlat coordinates from the site information table. We use the ‘site_unique’ field (unique combination of site and visit IDs) to link tables returned from the `get_ausplots` function:

```

my.fractional <- merge(my.fractional, my.ausplots.data$site.info, by = "site_unique")[,
  c("site_unique", "bare", "brown", "green", "NA.", "longitude", "latitude")]

```

```

my.fractional <- na.omit(my.fractional)

```

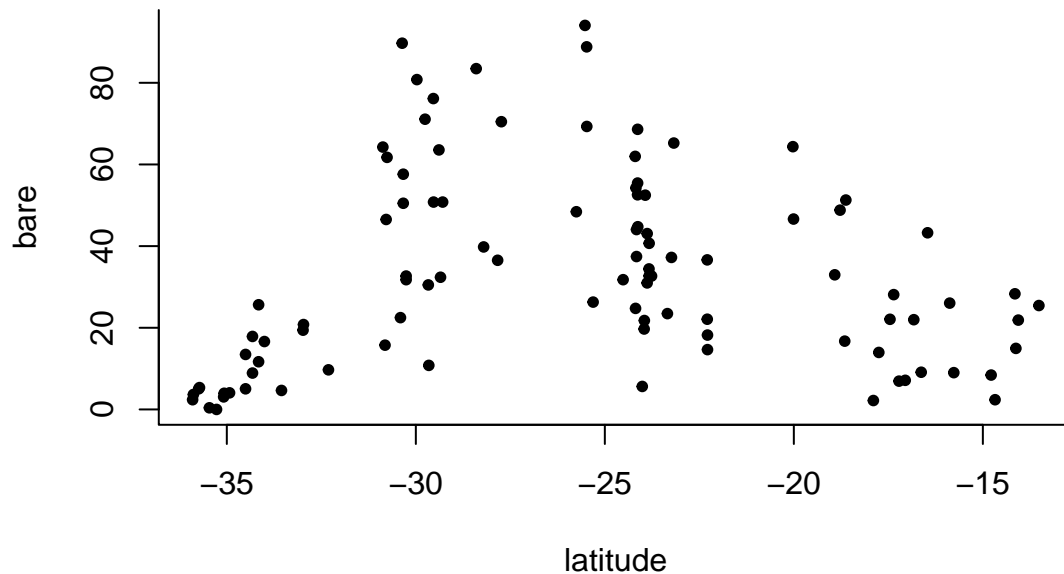
```

head(my.fractional)
#>          site_unique  bare brown green NA. longitude latitude
#> 1 NTAARP0003-58424 25.45 40.50 34.06 0 132.4571 -13.51697
#> 2 NTABRT0001-53616 18.22 33.27 48.51 0 133.2473 -22.28360
#> 3 NTABRT0003-53618 14.65 30.59 54.75 0 133.2699 -22.28358
#> 4 NTABRT0004-53619 22.10 31.42 46.48 0 133.6164 -22.28981
#> 5 NTABRT0005-53620 36.63 19.41 43.96 0 133.6121 -22.29108
#> 6 NTAFIN0002-53622  5.63 19.15 75.22 0 133.4343 -24.00942

```

Now we can plot out the continental relationship, e.g., between the proportion of bare ground with no kind of vegetation cover above and latitude.

```
plot(bare ~ latitude, data = my.fractional, pch = 20, bty = "l")
```



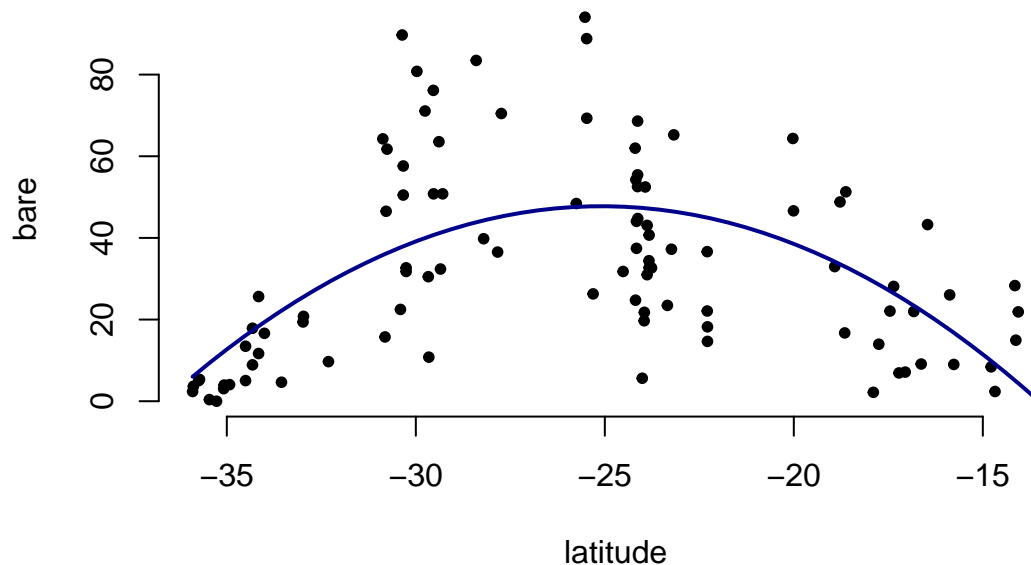
There appears to be a hump-backed relationship, with a higher proportion of bare ground in the arid inland at mid-latitudes. We can add a simple quadratic model to test/approximate this:

```
my.fractional$quadratic <- my.fractional$latitude^2

LM <- lm(bare ~ latitude + quadratic, data = my.fractional)
summary(LM)
#>
#> Call:
#> lm(formula = bare ~ latitude + quadratic, data = my.fractional)
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -41.719 -12.352  -3.899  10.287  51.866
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept) -176.71497    29.77860  -5.934 4.79e-08 ***
#> latitude     -17.89721     2.43854  -7.339 7.23e-11 ***
#> quadratic     -0.35673     0.04778  -7.467 3.94e-11 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 19.03 on 95 degrees of freedom
#> Multiple R-squared:  0.3712, Adjusted R-squared:  0.3579
#> F-statistic: 28.04 on 2 and 95 DF,  p-value: 2.696e-10

# generate predicted values for plotting:
MinMax <- c(min(my.fractional$latitude), max(my.fractional$latitude))
ND <- data.frame(latitude = seq(from = MinMax[1], to = MinMax[2], length.out = 50),
  quadratic = seq(from = MinMax[1], to = MinMax[2], length.out = 50)^2)
ND$predict <- predict(LM, newdata = ND)
```

```
#
plot(bare ~ latitude, data = my.fractional, pch = 20, bty = "n")
points(ND$latitude, ND$predict, type = "l", lwd = 2, col = "darkblue")
```



Example 2: Species by sites table

Aside from ‘gross’ values from plots such as fractional cover, many analyses in community ecology begin with species abundance information. With **ausplotsR** you can generate this easily from the more complex vegetation point-intercept data. The first step to work with species-level AusPlots data is to create a species occurrence matrix. The `species_table` function in the **ausplotsR** package can be used to create this type of matrix. This function takes a data frame of individual raw point-intercept hits (i.e. a `$veg.PI` data frame) generated using the `get_ausplots` function and returns a ‘species against sites’ matrix:

```
# The species_table function below can also take the '$veg.voucher' module as
# input, but 'm_kind='PA'' must be specified to get a sensible presence/absence
# output.
```

```
# The 'species_name' argument below specifies use of the 'standardised_name'
# field to identify species, which is based on herbarium_determination names
# (i.e., 'HD' option in species_name) matched to accepted scientific name
# according to a standard (http://www.worldfloraonline.org/).
```

```
my.sppBYsites <- species_table(my.ausplots.data$veg.PI, m_kind = "percent_cover",
  cover_type = "PFC", species_name = "SN")
```

```
# check the number of rows (plots) and columns (species) in the matrix
```

```
dim(my.sppBYsites)
```

```
#> [1] 391 1751
```

```
# look at the top left corner (as the matrix is large)
```

```
my.sppBYsites[1:5, 1:5]
```

```
#>           Abutilon Abutilon.fraseri Abutilon.halophilum Abutilon.hannii
#> NTAARP0001-58422           0           0           0           0
#> NTAARP0002-58423           0           0           0           0
#> NTAARP0003-58424           0           0           0           0
```

```
#> NTABRT0001-53616      0      0      0      0
#> NTABRT0002-53617      0      0      0      0
#>      Abutilon.leucopetalum
#> NTAARP0001-58422      0
#> NTAARP0002-58423      0
#> NTAARP0003-58424      0
#> NTABRT0001-53616      0
#> NTABRT0002-53617      0
```

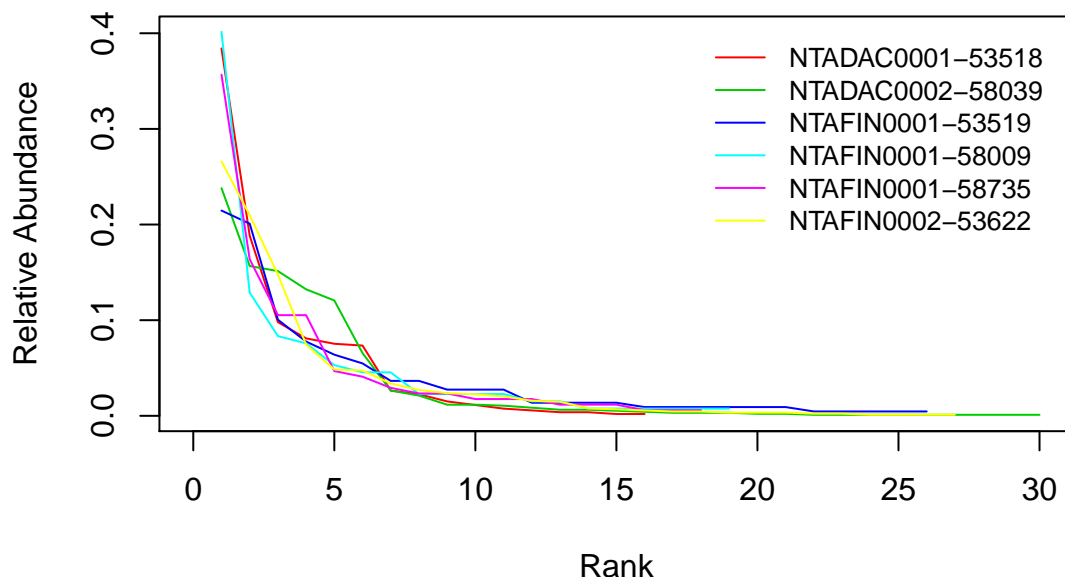
We can crudely pull out the 10 highest ranking species in terms of their percent cover cumulative across all plots they occur in:

```
rev(sort(colSums(my.sppBYsites)))[1:10]
#>      Triodia.basedowii      Aristida.holathera      Eucalyptus.obliqua
#>      651.4038      456.6403      359.5146
#>      Eulalia.aurea      Eucalyptus.baxteri      Triodia.bitextura
#>      340.4958      335.0025      325.2499
#>      Cenchrus.ciliaris      Triodia.pungens      Acacia.shirleyi
#>      306.7625      291.3861      282.0792
#> Schizachyrium.fragile
#>      264.3876
```

A simple example of downstream visualisation and analysis of species-level AusPlots data is Rank-Abundance Curves (also known as Whittaker Plots). Rank-Abundance Curves provide further information on species diversity. They provide a more complete picture than a single diversity index. Their x-axis represents the abundance rank (from most to least abundant) and in the y-axis the species relative abundance. Thus, they depict both Species Richness and Species Evenness (slope of the line that fits the rank; steep gradient indicates low evenness and a shallow gradient high evenness).

```
# Whittaker plots for some selected AusPlots:
goeveg::racurves(my.sppBYsites[10:15, ], bw = F)
```

Rank–abundance diagram

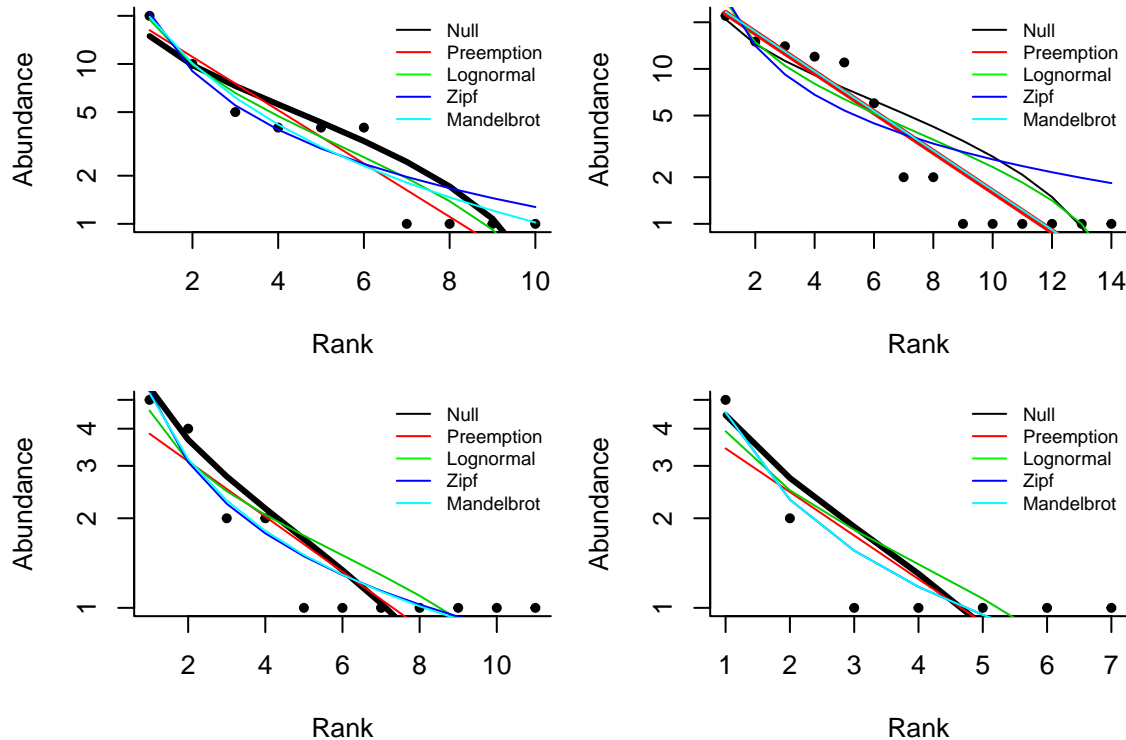


```
# And then with alternative relative abundance models fitted for communities
# individually:
```

```

par(mfrow = c(2, 2), mar = c(4, 4, 1, 1))
for (i in c(1:4)) {
  plot(vegan::radfit(round(my.sppBYsites[9 + i, ], digits = 0), log = "xy"), pch = 20,
       legend = FALSE, bty = "l")
  legend("topright", legend = c("Null", "Preemption", "Lognormal", "Zipf", "Mandelbrot"),
        lwd = rep(1, 5), col = c("black", "red", "green", "blue", "cyan"), cex = 0.7,
        bty = "n")
}

```



Example 3: Quick species lists

Perhaps you simply want to browse which plant species have been recorded in AusPlots, without all the associated raw data? Here, the `species_list` function is your friend:

```

# The species_list function is designed to take $veg.voucher as input but can
# also take $veg.PI

# print a list of genus_species-only records from selected plots:
species_list(subset(my.ausplots.data$veg.vouch, site_unique %in% unique(site_unique)[1:2]),
             grouping = "by_site", species_name = "GS")
#> $NTAMGD0002
#> [1] Abutilon hannii      Abutilon indicum
#> [3] Aristida latifolia   Astrebla elymoides
#> [5] Astrebla squarrosa   Boerhavia paludosa
#> [7] Brachyachne convergens Bulbostylis barbata
#> [9] Cleome viscosa       Crotalaria dissitiflora
#> [11] Crotalaria medicaginea Cyperus gilesii
#> [13] Cyperus iria         Dactyloctenium radulans
#> [15] Desmodium muelleri   Enneapogon polyphyllus
#> [17] Euphorbia            Fimbristylis dichotoma

```



```

#> [19] Gomphrena breviflora      Goodenia fascicularis
#> [21] Heliotropium conocarpum   Heliotropium plumosum
#> [23] Heliotropium tenuifolium Hibiscus
#> [25] Indigofera polygaloides  Ipomoea coptica
#> [27] Ipomoea lonchophylla     Iseilema fragile
#> [29] Kallstroemia pentandra   Neptunia monosperma
#> [31] Panicum laevinode       Paspalidium retiglume
#> [33] Phyllanthus lacerosus    Phyllanthus maderaspatensis
#> [35] Portulaca                Portulaca digyna
#> [37] Portulaca oleracea       Portulaca oligosperma
#> [39] Portulaca pilosa        Ptilotus exaltatus
#> [41] Ptilotus spicatus       Rhynchosia minima
#> [43] Salsola kali            Sclerolaena bicornis
#> [45] Sclerolaena lanicuspis  Sida argillacea
#> [47] Sida fibulifera         Sida spinosa
#> [49] Sorghum timorense       Sporobolus carolii
#> [51] Streptoglossa bubakii   Tragus australianus
#> [53] Trianthema triquetra    Uranthoecium truncatum
#>
#> $SATSTP0002
#> [1] Abutilon leucopetalum    Acacia ligulata
#> [3] Acacia victoriae        Aristida arida
#> [5] Casuarina pauper        Cenchrus ciliaris
#> [7] Convolvulus remotus    Dissocarpus paradoxus
#> [9] Dodonaea viscosa        Dysphania cristata
#> [11] Einadia nutans          Enchylaena tomentosa
#> [13] Enneapogon avenaceus    Enneapogon cylindricus
#> [15] Eragrostis dielsii      Eremophila sturtii
#> [17] Euphorbia drummondii    Lysiana exocarpi
#> [19] Maireana pyramidata     Nicotiana simulans
#> [21] Paractaenum novae-hollandiae Pimelea microcephala
#> [23] Rhagodia spinescens     Salsola kali
#> [25] Santalum lanceolatum    Sclerolaena obliquicuspis
#> [27] Senecio spanomerus      Senna artemisioides
#> [29] Sida ammophila          Sida fibulifera
#> [31] Solanum quadriloculatum Sonchus oleraceus
#> [33] Swainsona burkittii     Tetragonia eremaea
#> [35] Triraphis mollis

# overall species list ordered by family (for demonstration we print only part):
species_list(my.ausplots.data$veg.vouch, grouping = "collapse", species_name = "SN",
  append_family = TRUE)[1:20]
#> [1] Acanthaceae--Dicliptera armata
#> [2] Acanthaceae--Dipteracanthus australasicus
#> [3] Acanthaceae--Hygrophila ringens var. ringens
#> [4] Acanthaceae--Nelsonia canescens
#> [5] Acanthaceae--Rostellularia adscendens
#> [6] Aizoaceae--Carpobrotus rossii
#> [7] Aizoaceae--Carpobrotus virescens
#> [8] Aizoaceae--Disphyma clavellatum
#> [9] Aizoaceae--Gunnera
#> [10] Aizoaceae--Gunnera
#> [11] Aizoaceae--Gunnera

```

```
#> [12] Aizoaceae--Gunneriopsis quadrifida
#> [13] Aizoaceae--Gunneriopsis septifraga
#> [14] Aizoaceae--Gunneriopsis zygophylloides
#> [15] Aizoaceae--Mesembryanthemum crystallinum
#> [16] Aizoaceae--Mesembryanthemum nodiflorum
#> [17] Aizoaceae--Sarcocolla praecox
#> [18] Aizoaceae--Tetragonia
#> [19] Aizoaceae--Tetragonia eremaea
#> [20] Aizoaceae--Tetragonia implexicoma
```

Explore TERN AusPlots

In addition to the key site info and vegetation point-intercept modules introduced above, `get_ausplots` is your gateway to raw data modules for vegetation structural summaries, vegetation vouchers (covers the full species diversity observed at the plot and includes tissue sample details), basal wedge, and soils subsites, bulk density and pit/characterisation (including bulk and metagenomics soil samples).

References

- Blanco-Martin, B. (2019) Tutorial: Understanding and using the ‘ausplotsR’ package and AusPlots data. Terrestrial Ecology Research Network. Version 2019.04.0, April 2019. <https://github.com/ternaustralia/TERN-Data-Skills/>
- Sparrow, B., Foulkes, J., Wardle, G., Leitch, E., Caddy-Retali, S., van Leeuwen, S., Tokmakoff, A., Thurgate, N., Guerin, G.R. and Lowe, A.J. (2020) A vegetation and soil survey method for surveillance monitoring of rangeland environments. *Frontiers in Ecology and Evolution*, 8:157.