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

Greg Guerin & Bernardo Blanco-Martin

2020-08-11

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 subsetted 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.

```
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(veg.vouchers = FALSE, 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.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.

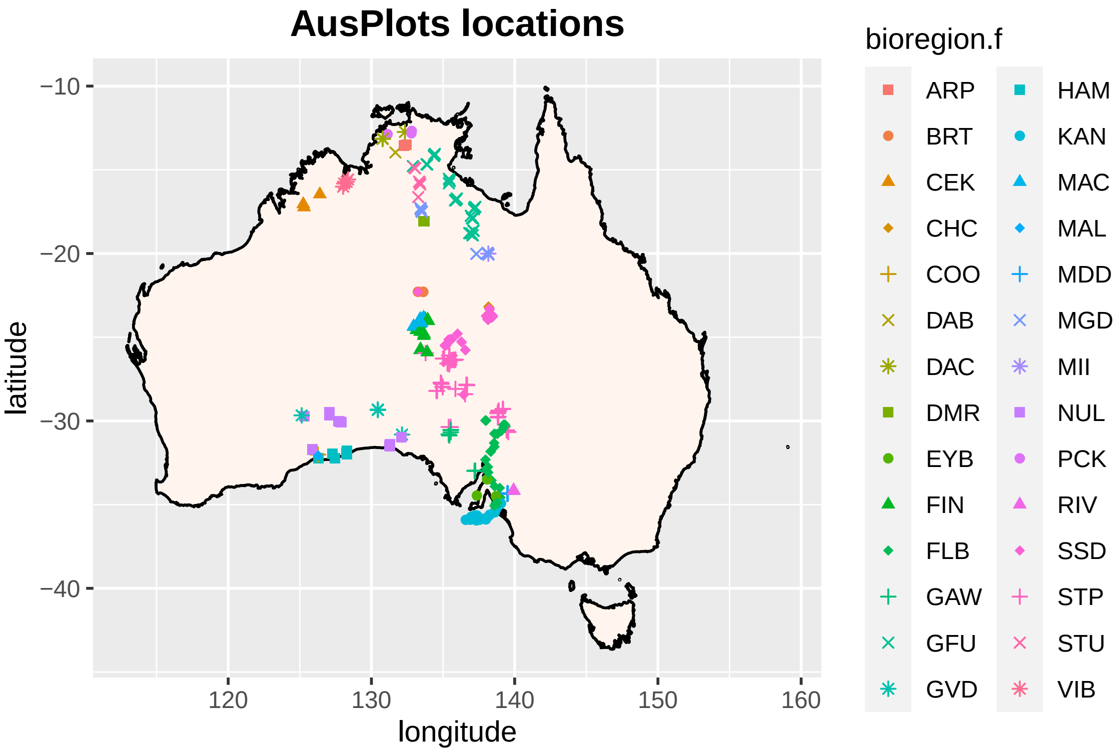
The 'site.info' table and can be used to identify, subset or group surveys in space and time, for example:

```
# 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 veg_barcode
#> 17         NTAMGD0002                  53466    E5-W5            2 NTA  006387
#> 20         NTAMGD0002                  53466    E5-W5            5 NTA  006345
#> 29         NTAMGD0002                  53466    E5-W5           15 NTA  006345
#> 30         NTAMGD0002                  53466    E5-W5           16 NTA  006317
#> 31         NTAMGD0002                  53466    E5-W5           17 NTA  006401
#> 33         NTAMGD0002                  53466    E5-W5           19 NTA  006387
#>    herbarium_determination substrate in_canopy_sky  dead   growth_form height hits_unique
#> 17       Sorghum timorense      Bare         FALSE FALSE Tussock grass   0.10     E5-W5 2
#> 20         Sida fibulifera      Bare         FALSE FALSE          Forb   0.05     E5-W5 5
#> 29         Sida fibulifera    Gravel         FALSE FALSE          Forb   0.10    E5-W5 15
#> 30        Iseilema fragile      Bare         FALSE FALSE Tussock grass   0.20    E5-W5 16
#> 31      Astrebla elymoides    Gravel         FALSE FALSE Tussock grass   0.20    E5-W5 17
#> 33       Sorghum timorense    Gravel         FALSE FALSE Tussock grass   0.10    E5-W5 19
#>         site_unique
#> 17 NTAMGD0002-53466
#> 20 NTAMGD0002-53466
#> 29 NTAMGD0002-53466
#> 30 NTAMGD0002-53466
#> 31 NTAMGD0002-53466
#> 33 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.
#> NTAARP0001-58422 NTAARP0001-58422  3.27 28.42 68.32 0.00
#> NTAARP0003-58424 NTAARP0003-58424 25.45 40.40 34.16 0.00
#> NTABRT0002-53617 NTABRT0002-53617 11.78 25.74 61.49 0.99
#> NTABRT0005-53620 NTABRT0005-53620 36.63 19.41 43.96 0.00
#> NTAFIN0002-53622 NTAFIN0002-53622  5.63 19.25 75.12 0.00
#> NTAFIN0002-58018 NTAFIN0002-58018 14.65 40.50 44.85 0.00
```

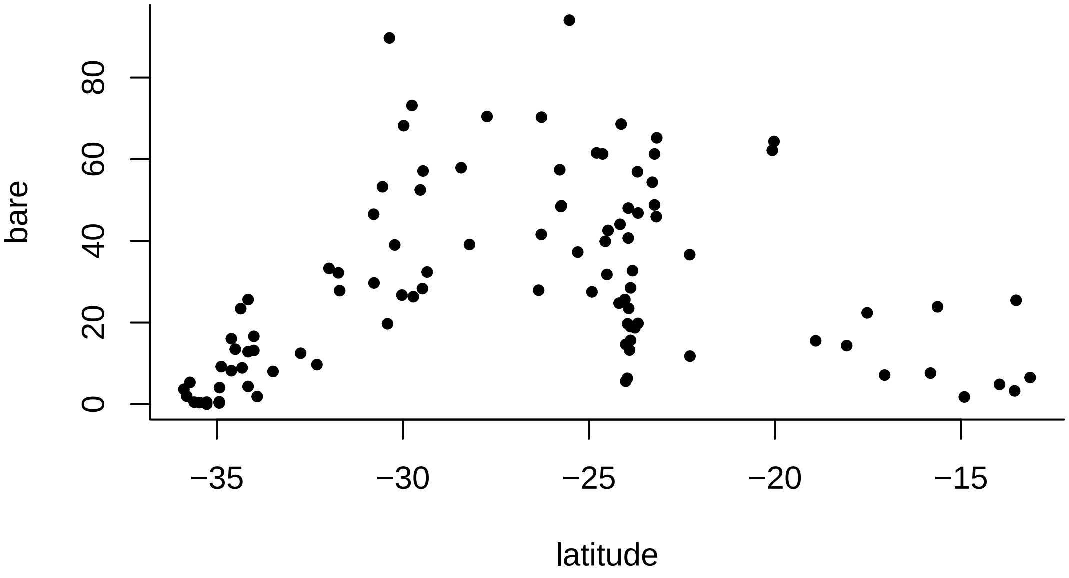
Next, we need to merge the fractional cover scores with longlat coordinates from the site information table.

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 NTAARP0001-58422  3.27 28.42 68.32 0.00  132.2701 -13.55729
#> 2 NTAARP0003-58424 25.45 40.40 34.16 0.00  132.4571 -13.51697
#> 3 NTABRT0002-53617 11.78 25.74 61.49 0.99  133.2506 -22.28367
#> 4 NTABRT0005-53620 36.63 19.41 43.96 0.00  133.6121 -22.29108
#> 5 NTAFIN0002-53622  5.63 19.25 75.12 0.00  133.4343 -24.00942
#> 6 NTAFIN0002-58018 14.65 40.50 44.85 0.00  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.

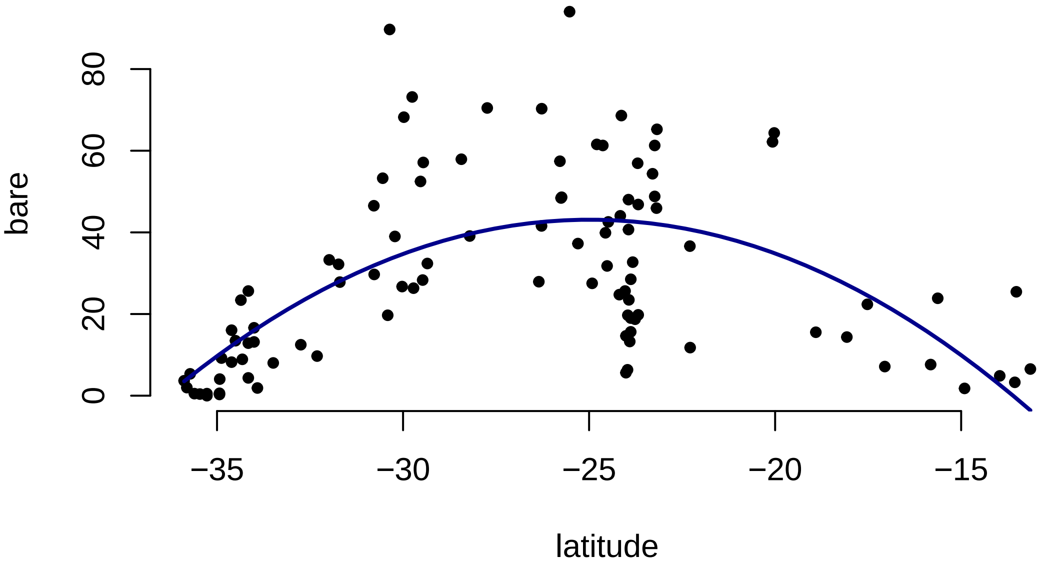
```
plot(bare ~ latitude, data = my.fractional, pch = 20, bty = "l")
```



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
#> -37.162 -10.619  -2.078   9.729  56.228
#>
```

```
#> Coefficients:
#>               Estimate Std. Error t value Pr(>|t|)
#> (Intercept) -164.71186   28.46523  -5.786 9.00e-08 ***
#> latitude     -16.63759    2.23952  -7.429 4.50e-11 ***
#> quadratic     -0.33299    0.04271  -7.796 7.65e-12 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 17.66 on 96 degrees of freedom
#> Multiple R-squared:  0.407,  Adjusted R-squared:  0.3946
#> F-statistic: 32.94 on 2 and 96 DF,  p-value: 1.279e-11

# 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:

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

# check the number of rows (plots) and columns (species) in the matrix
dim(my.sppBYsites)
```

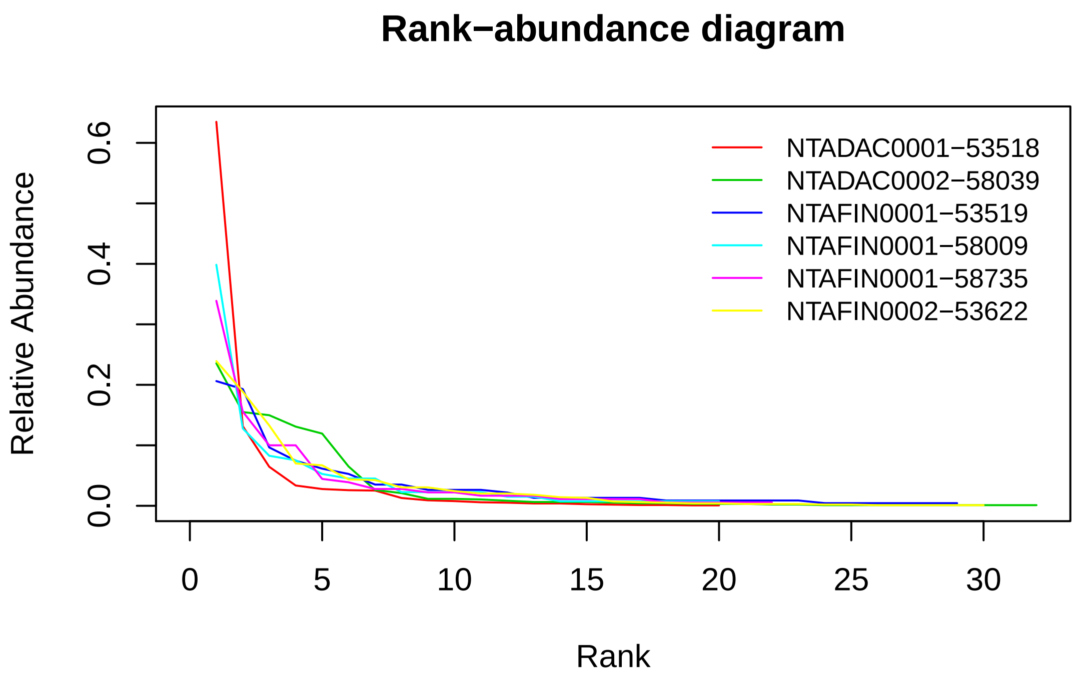
```
#> [1]  391 1956

# look at the top left corner (as the matrix is large)
my.sppBYsites[1:5, 1:5]
#>                     Abutilon.fraseri Abutilon.halophilum Abutilon.hannii
#> NTAARP0001-58422                0                   0               0
#> NTAARP0002-58423                0                   0               0
#> NTAARP0003-58424                0                   0               0
#> NTABRT0001-53616                0                   0               0
#> NTABRT0002-53617                0                   0               0
#>                     Abutilon.hannii.subsp..prostrate..p.k.latz.427. Abutilon.leucopetalum
#> NTAARP0001-58422                                               0                     0
#> NTAARP0002-58423                                               0                     0
#> NTAARP0003-58424                                               0                     0
#> NTABRT0001-53616                                               0                     0
#> NTABRT0002-53617                                               0                     0
```

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

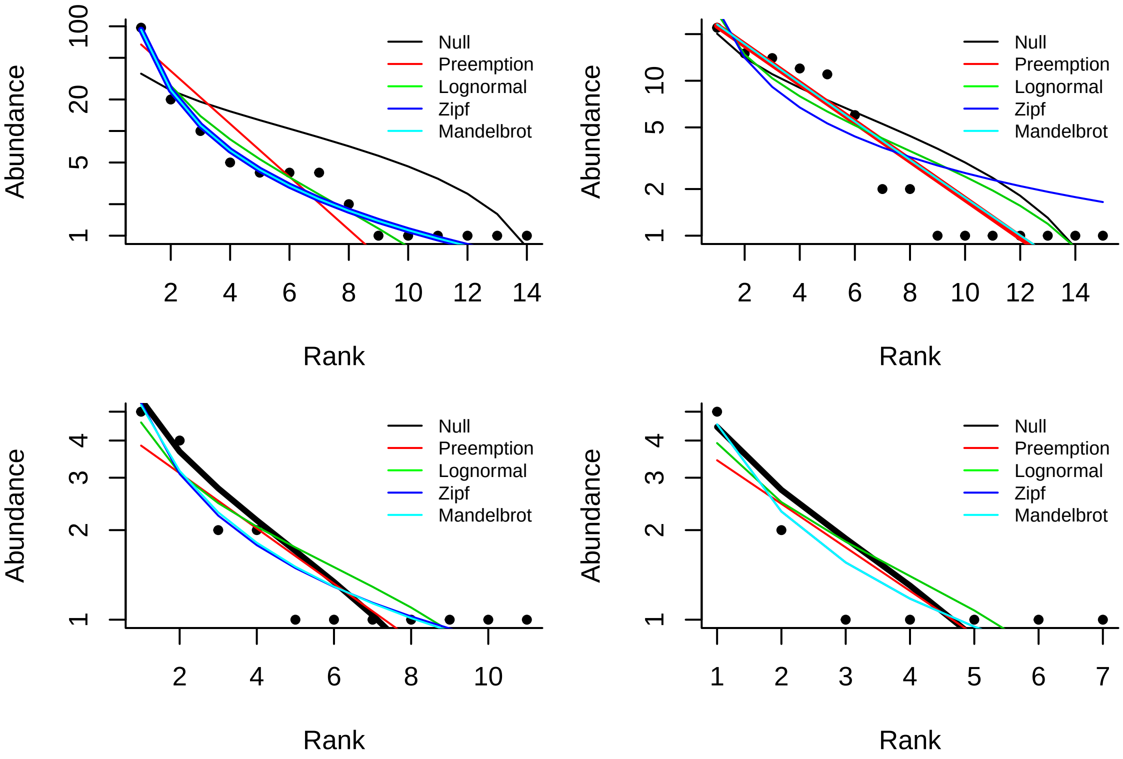
```
rev(sort(colSums(my.sppBYsites)))[1:10]
#>     Triodia.basedowii    Aristida.holathera    Eucalyptus.obliqua          Eulalia.aurea
#>              651.4038              399.9520              359.5146               340.4958
#>    Eucalyptus.baxteri     Triodia.bitextura     Cenchrus.ciliaris        Triodia.pungens
#>              335.0025              325.2499              306.7625               291.3861
#>       Acacia.shirleyi Schizachyrium.fragile
#>              282.0792              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)
```



```
# 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")
}
```



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-Retalic, 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.