

# Package ‘soilprofile’

July 2, 2014

**Type** Package

**Title** A package to consistently represent soil properties along a soil profile

**Version** 1.0

**Date** 2013-02-25

**Author** Gianluca Filippa

**Maintainer** Gianluca Filippa <gian.filippa@gmail.com>

**Description** This package provides functions to graphically represent soil properties. Morphological data gathered in the field such as horizon boundaries, root abundance and dimensions, skeletal shape, abundance and dimension as well as meaningful soil color may be represented via the plot function. A lattice-based plot.element function has been designed to represent depth function of a given variable.

**Depends** R (>= 2.14.1), aqp, lattice, munsell, splancs, methods

**LazyLoad** yes

**LazyData** yes

**License** GPL-2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-08-24 00:42:57

## R topics documented:

soilprofile-package . . . . .	2
build.profile . . . . .	3
channer . . . . .	5
depths . . . . .	5
eplot . . . . .	6

example . . . . .	7
example.data . . . . .	8
head.profile.data.frame . . . . .	8
munsell_to_rgb . . . . .	9
plot.profile.data.frame . . . . .	10
plot_profile . . . . .	11
print.profile.data.frame . . . . .	14
retrieve.horizons . . . . .	15
retrieve.roots . . . . .	15
retrieve.skeletal . . . . .	16
roots . . . . .	17
root_unit . . . . .	17
skeletal . . . . .	18
subangular . . . . .	18
subcircle . . . . .	19
summary.profile.data.frame . . . . .	19
tail.profile.data.frame . . . . .	20

## Index 21

---

soilprofile-package    *A package for consistently drawing soil profiles and their properties*

---

## Description

This package provides functions to graphically represent soil properties. Morphological data gathered in the field such as horizon boundaries, root abundance and dimensions, skeletal shape, abundance and dimension as well as meaningful soil munsell color may be represented via the plot function. A lattice-based plot.element function has been designed to represent depth function of a given variable.

## Details

Package: soilprofile  
 Type: Package  
 Version: 1.0  
 Date: 2013-02-25  
 License: GPL-2

## Author(s)

Gianluca Filippa Maintainer: Gianluca Filippa <gian.filippa@gmail.com>

## References

Literature or other references for background information ~~~

## Examples

```
plot(example, existing_data=example.data, random=FALSE)
eplot(example, 'Corg', type='b', col='black')
```

---

build.profile	<i>Converts a data.frame into a profile.data.frame</i>
---------------	--

---

## Description

This function simply converts a data.frame into an object of class profile.data.frame. A table can be provided with suitable names (as in the arguments of the function) or single vectors.

## Usage

```
build.profile(table, Profile = NULL, name = NULL, depth = NULL, col = NULL,
skel_dim = NULL, skel_ab = NULL, type = NULL, root_ab = NULL,
root_dim = NULL, orientation = NULL)
```

## Arguments

table	A data.frame containing the variables needed to draw a soil profile. If a variable with the required name is missing, the function looks for it in subsequent arguments (with a warning).
Profile	A vector containing coding of profiles. Multiple profiles are allowed in the same dataframe. Profiles may be in any format.
name	A vector containing the code for each single horizon. Horizons are allowed in usual form (e.g. A1, Bw, etc.) or as transition horizons (O/A) or as 'cumulative' horizons as it usually occurs for O horizons (Oi-Oe-Oa).
depth	A vector of depths. Depth must be in the form of upper_boundary - lower_boundary. Irregular boundaries are allowed and may be indicated in the form lower1/lower2. See examples for details.
col	A vector of colors in the form hue value/chroma. Vectors are passed to 'as.character' for conversion. Hue and chroma must be strictly separated by a space, whereas chroma and value by a '/'. Non integer numbers for hue and chroma are rounded.
skel_dim	A numeric vector of skeletal dimensions in cm.
skel_ab	A numeric vector of skeletal abundance in the range 0-1.
type	A character vector indicating the type of skeletal unit. Three types of skeletal shape are allowed. 'subangular', 'subcircle' and 'channer'. The basic shape is loaded from a matrix of coordinates.

root_ab	A character vector giving the root abundance expressed qualitatively. Available abundances are 'absent', 'few', 'common', 'many'. Those qualitative indices are chosen according to the international classification systems.
root_dim	A numeric vector providing the dimension of roots given in cm.
orientation	A character vector to be chosen between 'v' and 'h' to indicate main orientation of the root system, vertical and horizontal, respectively.

### Details

This function converts a data.frame in an object of class 'profile.data.frame'. This class has dedicated methods such as summary, head, tail, plot, etc. See examples for more details. In the easiest case, a data.frame can be passed to this function, which already has proper column names (as in the arguments of the function) of a 'profile.data.frame'. If not, each vector must be provided in the proper argument of the function. Warnings are set in order to facilitate this process of conversion. Additional data (e.g. chemical or physical data associated to the profiles' horizons) are arranged with unchanged names after the 10 required columns.

### Value

An object of class 'profile.data.frame'. This object is actually a list, and only few basic methods have been defined to explore it (eg. 'summary', 'head', 'tail', 'plot', 'eplot'). All other functions will treat this class as a list and therefore lead to meaningless results.

### Author(s)

Gianluca Filippa

### See Also

'plot.profile.data.frame', 'plot.element'

### Examples

```
##rename
tmp <- example
## back to class data.frame
class(tmp) <- 'data.frame'
## back again to the 'profile.data.frame' class
profile.data <- build.profile(tmp)
## if the table is not complete, we remove the Profile column
tmp2 <- tmp[,-1]
##running the following will return a warning
profile.data <- build.profile(tmp2)
##and Profile column is NA
head(profile.data)
##
Profile <- tmp[,1]
##check information in output when we add the Profile vector
profile.data <- build.profile(tmp2, Profile=Profile)
```

---

channer	<i>Dataset of coordinates for a channer-shaped skeleton unit</i>
---------	--

---

**Description**

Dataset of coordinates for a channer-shaped skeleton unit

**Usage**

```
data(channer)
```

**Format**

A matrix of coordinates

**Details**

This dataset is internally loaded in the function 'plot.profile.data.frame' to draw basic 'channer' shaped stone units.

**Examples**

```
plot(channer, type='n', axes=FALSE)
polygon(channer)
```

---

depths	<i>Converts depths into suitable mean points for each horizon in a soil profile</i>
--------	---

---

**Description**

This function is internally called in 'build.profile', but can be run by itself when you want to plot a depth function of given soil properties.

**Usage**

```
depths(depths, horizon.names)
```

**Arguments**

**depths** A vector of depths in the form upper\_boundary - lower\_boundary or in the form upper\_boundary - lower\_boundary1/lower\_boundary2 in case of irregular boundaries. Conversion is simply an average between upper and lower boundary. See examples for details.

**horizon.names** A vector of names, same as 'names' option in build.profile().

**Value**

A dataframe with a vector containing the given mean point for each horizon and a vector with horizon names.

**Author(s)**

Gianluca Filippa

**See Also**

'plot.profile.data.frame', 'eplot'

**Examples**

```
##extract mean point
tmp <- depths(example$depth, example$name)
##compare mid points and original depths
data.frame(example$depth, tmp[,1])
```

---

eplot

*A lattice-based plot of elements on multiple panels*

---

**Description**

This function is used for plotting depth functions of given elements in one or more soil profile(s)

**Usage**

```
eplot(data, element, panel = NULL, groups = NULL, ...)
```

**Arguments**

data	An object of class 'profile.data.frame'
element	A string giving the element as in column name of data
panel	A vector of grouping factor for the conditioning variables (which define the subsets plotted in different panels) to be used in the plot. If NULL it defaults to the column named Profile, and plots elements as defined in 'element' subsetted for different profiles.
groups	An additional subgroup which controls colors of the elements plotted in each panel.
...	Other arguments as in 'xyplot' function in package 'lattice'.

**Details**

This function plots element in 'element' according to the subsetting provided in 'panel', and 'groups'. It is provided to rapidly obtain a depth function of a given variable, subsetted by profile in case of multiple-profile datasets.

**Value**

The high-level functions documented here, as well as other high-level Lattice functions, return an object of class 'trellis'. See ?xyplot for more details.

**Author(s)**

Gianluca Filippa

**See Also**

plot.profile.data.frame

**Examples**

```
## plot of organic C depth functions in 5 profiles
eplot(example, 'Corg', col='black', type='b', main='Example',
       xlab='Organic C (percent)', ylab='Depth')

## different colors according to different soil profiles
eplot(example, 'Corg', type='b', main='Example',
       xlab='Organic C (%)', ylab='Depth', groups=example$Profile,
       panel=1, col=palette()[1:4])

## an example to add a legend with trellis (lattice) commands
trellis.focus("toplevel") ## has coordinate system [0,1] x [0,1]
panel.text(0.65, 0.63, "Site", cex = 0.8, font = 2, pos=4)
panel.text(0.65, 0.59, "P1", cex = 0.8, pos=4)
panel.text(0.65, 0.56, "SG2", cex = 0.8, pos=4, col=palette()[2])
panel.text(0.65, 0.53, "SG5", cex = 0.8, pos=4, col=palette()[3])
panel.text(0.65, 0.50, "SG7", cex = 0.8, pos=4, col=palette()[4])
trellis.unfocus()
```

---

example

*An example of class 'profile.data.frame'*

---

**Description**

This example is from a recent publication of soil properties gathered in the field. For more info see D'Amico et al. 2013. The dataset consists of four profiles along a glacier foreland chronosequence in the North-western Alps.

**Usage**

```
data(example)
```

**Format**

The format is actually a list but the underlying structure is a data.frame

**References**

D'amico et al. 2013 (submitted)

**Examples**

```
summary(example)
```

---

```
example.data
```

*Data generated from dataframe 'example' to plot profiles*

---

**Description**

This example is from a recent publication of soil properties gathered in the field. For more info see D'Amico et al. 2013. The dataset consists of four profiles along a glacier foreland chronosequence in the North-western Alps.

**Usage**

```
data(example.data)
```

**Format**

A complex list containing all numbers needed to plot soil profiles as described in example

**Examples**

```
## Not run: plot(example, random=FALSE, existing_data=example.data)
```

---

```
head.profile.data.frame
```

*Head of an object of class 'profile.data.frame'*

---

**Description**

As the generic function 'head'

**Usage**

```
## S3 method for class 'profile.data.frame'
head(x, ...)
```

**Arguments**

x                    An object of class 'profile.data.frame'  
 ...                  Same arguments as the generic function 'head'



**Value**

An object (usually) like 'x' but generally smaller.

**See Also**

tail.profile.data.frame

**Examples**

```
head(example)
```

---

munsell_to_rgb	<i>Converts munsell colors to RGB colors</i>
----------------	--

---

**Description**

This function converts a vector of munsell colors in the form hue value/chroma into an RGB R color. Vectors are passed to 'as.character' for conversion. Hue and chroma must be strictly separated by a space, whereas chroma and value by a '/'. It is internally called in the 'plot.profile.data.frame' function.

**Usage**

```
munsell_to_rgb(color, name)
```

**Arguments**

color	A string or vector containing Munsell color(s) in the form Hue value/chroma.
name	This argument is taken in the workflow of the 'plot.data.frame' function, with the purpose to give the horizon name in case of warning.

**Details**

The function separates Hue, value and chroma by calling 'substr' and then calls the function munsell2rgb aqp for the conversion into R colors. If the color is not existing (be either because of wrong separation between Hue value and chroma, or unexisting munsell color), it defaults to '#39302CFF' and returns a warning.

**Value**

A vector of R colors

**Author(s)**

Gianluca Filippa

## Examples

```
##a 5Y 5/3 (dark brown) boxplot  
boxplot(1:10, col=munsell_to_rgb(example[[6]][1]), main='A 5Y 5/3 (dark brown) boxplot')
```

---

plot.profile.data.frame

*The main plotting function for a soil profile*

---

## Description

A function that draws a soil profile based on simple field data.

## Usage

```
## S3 method for class 'profile.data.frame'  
plot(x, y, ...)
```

## Arguments

x	An object of class 'profile.data.frame'
y	Unused argument (only for matching the arguments of generic plot function)
...	All arguments allowed in 'plot_profile' function. See details.

## Details

This function is a generic 'plot' function for objects of class 'profile.data.frame'. For details of the arguments see 'plot\_profile'.

## Value

A complex list of all numeric values necessary to plot one (or more) soil profile(s). The plot can be assigned and used to plot the same soil profile(s) more than once. See arguments 'random' and 'existing data' in 'plot\_profile' for details.

## Author(s)

Gianluca Filippa

## Examples

```
## Not run:  
plot(example, random=FALSE, existing_data=example.data)  
plot(example)  
  
## End(Not run)
```

---

plot_profile	<i>The function which is called from the generic 'plot.profile.data.frame'</i>
--------------	--

---

## Description

This is the core function for plotting soil profile field data

## Usage

```
plot_profile(data, bottom = NULL, names = TRUE, names.col = "white",
background = "munsell", plot.roots = TRUE, plot.skeletal = TRUE,
random = TRUE, existing_data = NULL, horizon.border = NA, order = FALSE,
width = 480, element = FALSE, element.col = "black", element.legend =
FALSE, element.lims = FALSE, element.lab = FALSE, element.type = "b",
element.pch = 1, element.lty = 1, xax.log = FALSE, legend.labs = FALSE,
legend.pos = "bottom")
```

## Arguments

data	An object of class 'profile.data.frame' as generated by the 'build.profile' function.
bottom	A positive numeric value which controls the lower boundary of the plot. If not provided it defaults to the lower limit of the profile, or the lower limit of the deepest profile in case of multiple profiles.
names	A logical value indicating whether the name of the horizons should be plotted (the default) or not in the soil profile.
names.col	The color of the horizon names.
background	The color background for the horizons. If background='munsell' (the default), color is determined by a call to the 'munsell_to_rgb' function. See the function for details on the conversion.
plot.roots	A logical value indicating whether the roots should be plotted or not.
plot.skeletal	A logical value indicating whether the skeletal should be plotted or not.
random	A logical value indicating whether data for plotting elements (roots and skeletal) in the profile should be randomly generated or not. It defaults to TRUE and is required to be TRUE when you plot a profile for the first time. If you assign this first plot to an object, you can use such object in the following argument ('existing_data') to draw the same profile you have generated previously. Since the shapes are by default randomly generated, you can run 'plot.profile.data.frame' more than once until you get the soil profile representation you like most. Remember to always assign a plot to an object so that you can retrieve it. See examples for details.
existing_data	If random=FALSE, this is a complex list of elements for drawing the same soil profile(s) previously generated. See examples.
horizon.border	A color string. The default is NA, so that no horizon border is plotted.

order	A vector of names that match the profile names in your dataset. This is used for multiple profile plots. The default is alphabetical order, so if you want a customized order use this argument.
width	An integer value ranging approximately 150-1400. This number defaults to 480. It controls the 'layout' of multiple panels in a single figure. The default is optimized for plotting 3-4 profiles in a single figure. Lower widths are required for lower number of profiles and the other way around. Optimal width of plots is achieved by using the same width as argument of the plot function and as argument for width of your favorite graphical device. See examples.
element	A character vector of one or more parameters (chemical elements, etc.) to be plotted within the profile. Names must match colnames of your data. It defaults to FALSE, no element is plotted.
element.col	A character vector of colors for plotting elements.
element.legend	A logical value for plotting the legend.
element.lims	As for 'xlims' in plot, to set a unique limit for multiple elements in plot.
element.lab	A label for the x axis if element!=FALSE
element.type	One between 'n', 'b', 'l' as for 'plot', to define line type for element(s)
element.pch	As in 'pch' for 'plot'.
element.lty	As in 'lty', for 'plot'.
xax.log	A logical value for logarithmic x axis
legend.labs	if element.legend=TRUE, provide labs for legend
legend.pos	Defaults to 'bottom'. The location may be specified by setting a single keyword from the list 'bottomright', 'bottom', 'bottomleft', 'left', 'topleft', 'top', 'topright', 'right' and 'center' as in 'legend'. However, the legend position defaults to bottom and is optimized for being set in the panel with the shallower profile. So 'bottom' will work in most cases.

## Details

This function is the core of the package and is written for plotting basic soil properties usually gathered in the field in a semi-quantitative way, for consistent representation of soil profile sketches. Those sketches are common in soil science publications and provide immediate feeling of what a given profile, or even better a sequence of soils look like morphologically. This function makes it possible to draw such profiles in few seconds and in a semi-quantitative way. Horizon boundaries are drawn according to the given depths. Irregular boundaries in the form of e.g. '0-12/23' cm are allowed, where it is meant that the lower boundary of such horizon fluctuates between 12 and 23 cm. Stones (i.e. soil skeletal) are plotted within a soil horizon in three different shapes ('chanter', 'subangular', 'subcircle'), in whatever dimension expressed in cm, and with the abundance provided between 0 and 1. Abundance is computed exactly on an area basis for each horizon. Roots are drawn following the same approach as skeletal, except that root abundance is provided qualitatively, as it is a common practice in soil profile description. The color of a given horizon may be an RGB color that exactly reproduces the munsell color of the soil horizon.

**Value**

A complex list of all numeric values used in the plot are returned invisibly. The function can be however assigned for plotting the same soil profile more than once. See arguments 'random' and 'existing\_data' for more details.

**Author(s)**

Gianluca Filippa

**See Also**

'eplot', 'build.profile', 'depths'

**Examples**

```
## basic plot, assigned to an object
## Not run:
tmp2 <- plot(example)

## End(Not run)

## a different plot (data for plotting are randomly generated each time,
## unless you assign and recycle it, like in following examples)
## Not run:
plot(example)

## End(Not run)

## use existing_data and random=FALSE to get the same plot (if assigned)

## Not run:
plot(example, random=FALSE, existing_data=example.data)

## End(Not run)

## esclude roots and skeletal from a plot

## Not run:
plot(example, random=FALSE, existing_data=example.data, plot.roots=FALSE,
plot.skeletal=FALSE)

## End(Not run)

## a complex plot with an element added to the basic soil structure and a
## legend, and an overview of options

## Not run:
plot(example, random=FALSE, existing_data=example.data, horizon.border='blue',
width=500, plot.roots=FALSE, plot.skeletal=FALSE, element='Corg',
element.lim=c(0, 27), element.col='black', names=FALSE, background='white',
element.lab='Organic carbon (percent)', element.legend=TRUE,
legend.labs='TOC')
```

```
## End(Not run)

## an optimal plotting with its graphical device, note the use of width
## both in the device and within the plot.
## Not run:
png('myprofile.png', height=800, width=1200, pointsize=28)
plot(example, random=FALSE, existing_data=example.data, horizon.border='blue',
width=1200, plot.roots=FALSE, plot.skeletal=FALSE, element='Corg',
element.lim=c(0, 27), element.col='black', names=FALSE, background='white',
element.lab='Organic carbon (percent)', element.legend=TRUE,
legend.labs='TOC')
dev.off()

## End(Not run)
```

---

print.profile.data.frame

*Generic print function for profile.data.frame*

---

## Description

Prints an object of class 'profile.data.frame'

## Usage

```
## S3 method for class 'profile.data.frame'
print(x, ...)
```

## Arguments

x	An object of class 'profile.data.frame'
...	Any argument of the 'print' function

## Details

Prints an object of class 'profile.data.frame'

## Value

The profile.data.frame.object is printed as a data.frame

## Author(s)

Gianluca Filippa

## See Also

'plot.profile.data.frame', 'build.profile', 'head.profile.data.frame', 'summary.profile.data.frame'

**Examples**

```
print(example)
example
```

---

retrieve.horizons	<i>Recycles all produced data for drawing one or more profile(s) horizons</i>
-------------------	---

---

**Description**

Internally called by 'plot.profile.dataframe' when random=F

**Usage**

```
retrieve.horizons(profile_data, a, col, lwd, horizon.border = horizon.border)
```

**Arguments**

profile_data	Recycled arguments in 'plot.profile.data.frame'
a	Recycled arguments in 'plot.profile.data.frame'
col	Recycled arguments in 'plot.profile.data.frame'
lwd	Recycled arguments in 'plot.profile.data.frame'
horizon.border	Recycled arguments in 'plot.profile.data.frame'

**Details**

This function is internally called in plot.profile.data.frame when random=F

**Author(s)**

Gianluca Filippa

---

retrieve.roots	<i>Recycles roots features for drawing one or more profile(s)</i>
----------------	---

---

**Description**

Internally called by 'plot.profile.dataframe' when random=F

**Usage**

```
retrieve.roots(profile_data, a)
```

**Arguments**

profile\_data Recycled arguments in 'plot.profile.data.frame'  
a Recycled arguments in 'plot.profile.data.frame'

**Details**

This function is internally called in plot.profile.data.frame when random=F

**Author(s)**

Gianluca Filippa

---

retrieve.skeletal      *Recycles all produced data for drawing one or more profile(s) horizons*

---

**Description**

Internally called by 'plot.profile.dataframe' when random=F

**Usage**

```
retrieve.skeletal(profile_data, a, col)
```

**Arguments**

profile\_data Recycled arguments in 'plot.profile.data.frame'  
a Recycled arguments in 'plot.profile.data.frame'  
col Recycled arguments in 'plot.profile.data.frame'

**Details**

This function is internally called in plot.profile.data.frame when random=F

**Author(s)**

Gianluca Filippa



---

roots	<i>A function for drawing roots</i>
-------	-------------------------------------

---

**Description**

Internally called in 'plot.profile.data.frame', it draws roots in a soil profile

**Usage**

```
roots(horizon, root.abundance, root.dimension, orientation = "h")
```

**Arguments**

horizon  
root.abundance  
root.dimension  
orientation

**Details**

Details are available in 'plot.profile.data.frame' or 'plot\_profile'

**Author(s)**

Gianluca Filippa

---

root_unit	<i>A list of tree matrices of coordinates for root units</i>
-----------	--

---

**Description**

A list of tree matrices of coordinates for root units

**Usage**

```
data(root_unit)
```

**Format**

The format is a list of 4 matrices of coordinates

**Examples**

```
par(mfrow=c(1,3), mar=rep(2,4))
plot(root_unit[[1]], type='l', axes=FALSE)
plot(root_unit[[2]], type='l', axes=FALSE)
plot(root_unit[[3]], type='l', axes=FALSE)
par(mfrow=c(1,1))
```

---

skeletal	<i>Plot skeletal, internal function</i>
----------	---

---

**Description**

Plot skeletal, internal function

**Usage**

```
skeletal(horizon, clast_dimension, type, abundance, col)
```

**Arguments**

horizon	The given horizon
clast_dimension	In cm
type	A character value to be chosen between 'subangular', 'chanter', 'subcircle'
abundance	Ranges between 0-1
col	Recycled

**Details**

This function is internally called in 'plot.profile.data.frame'. See this function for details.

**Author(s)**

Gianluca Filippa

---

subangular	<i>Dataset of coordinates for a subangula-shaped skeleton unit</i>
------------	--

---

**Description**

Dataset of coordinates for a subangula-shaped skeleton unit

**Usage**

```
data(subangular)
```

**Format**

A matrix of coordinates

**Examples**

```
plot(subangular, type='n', axes=FALSE)
polygon(subangular)
```

---

`subcircle`*Dataset of coordinates for a subcircle-shaped skeleton unit*

---

**Description**

Dataset of coordinates for a subcircle-shaped skeleton unit

**Usage**

```
data(subcircle)
```

**Format**

A matrix of coordinates

**Examples**

```
plot(subcircle, type='n', axes=FALSE)
polygon(subcircle)
```

---

`summary.profile.data.frame`*Generic function to summarize an object of class 'profile.data.frame'*

---

**Description**

It summarizes an object of class 'profile.data.frame'

**Usage**

```
## S3 method for class 'profile.data.frame'
summary(object, ...)
```

**Arguments**

<code>object</code>	An object of class 'profile.data.frame'
<code>...</code>	All arguments for function 'summary'

**Details**

A summary function for an object of class 'profile.data.frame'

**Author(s)**

Gianluca Filippa

**See Also**

'head.profile.data.frame'

**Examples**

```
summary(example)
```

---

```
tail.profile.data.frame  
      generic function
```

---

**Description**

Same as tail generic function, see ?tail

**Usage**

```
## S3 method for class 'profile.data.frame'  
tail(x, ...)
```

**Arguments**

x  
...

**Examples**

```
tail(example)
```

# Index

## \*Topic **Format data**

build.profile, 3  
depths, 5

## \*Topic **Generic functions**

head.profile.data.frame, 8  
print.profile.data.frame, 14  
summary.profile.data.frame, 19  
tail.profile.data.frame, 20

## \*Topic **Internal functions**

munsell\_to\_rgb, 9  
retrieve.horizons, 15  
retrieve.roots, 15  
retrieve.skeletal, 16  
roots, 17  
skeletal, 18

## \*Topic **Plotting functions**

eplot, 6  
plot.profile.data.frame, 10  
plot\_profile, 11

## \*Topic **datasets**

channer, 5  
example, 7  
example.data, 8  
root\_unit, 17  
subangular, 18  
subcircle, 19

## \*Topic **package**

soilprofile-package, 2

build.profile, 3

channer, 5

depths, 5

eplot, 6

example, 7

example.data, 8

head.profile.data.frame, 8

munsell\_to\_rgb, 9

plot.profile.data.frame, 10

plot\_profile, 11

print.profile.data.frame, 14

retrieve.horizons, 15

retrieve.roots, 15

retrieve.skeletal, 16

root\_unit, 17

roots, 17

skeletal, 18

soilprofile (soilprofile-package), 2

soilprofile-package, 2

subangular, 18

subcircle, 19

summary.profile.data.frame, 19

tail.profile.data.frame, 20