

The “ArealSampling” Class

Jeffrey H. Gove*
Research Forester
USDA Forest Service
Northern Research Station
271 Mast Road
Durham, New Hampshire 03824 USA
e-mail: jgove@fs.fed.us or e-mail: jhgove@unh.edu

Monday 20th December, 2010
2:12pm

Contents

	2.1	Class slots	3
	3	The “circularPlot” Class	3
	3.1	“circularPlot” Class slots	4
1	1	Introduction	
2	2	The “ArealSampling” Class	
	3.2	Object creation	4
	3.3	Plotting the object	5

1 Introduction

The “ArealSampling” class is a virtual class that is used as a basis for each of the possible different areal sampling methods we use in forestry, whether for down logs or standing trees. For each of the subclasses, relevant information defining the sampling method should be given that will allow the computation of its associated inclusion zone later in the “InclusionZone” class. Because most areal sampling methods also depend on the attributes of the “Stem” subclass that represents it (i.e., the inclusion zone for PPS methods especially are of this form), most subclasses will not have any “SpatialPolygons” slot available for rendering the object graphically. One obvious exception is with fixed-radius plots under, e.g., the ‘standup’ method (Gove and Van Deusen, 2011). In addition, since ‘standup,’ ‘chainsaw,’ and ‘sausage’ are simply protocol differences within the fixed-area circular plot method of sampling, we do not differentiate them here, but wait until the “InclusionZone” class to make that distinction.

An overview of the “ArealSampling” class structure is presented in Figure [1](#). At this point it is uncertain whether there will be a division between standing tree and down log methods. Such a

*Phone: (603) 868-7667; Fax: (603) 868-7604.

division is somewhat artificial as some of the methods, such as circular plot sampling, can be used on both, and it would be redundant to have them defined twice. But we can always have a “joint” subclass, encompassing these methods, or just let them hang by themselves as necessary. At this point in the design, none of these changes should impact what is already completed. Furthermore, it

The ArealSampling Class

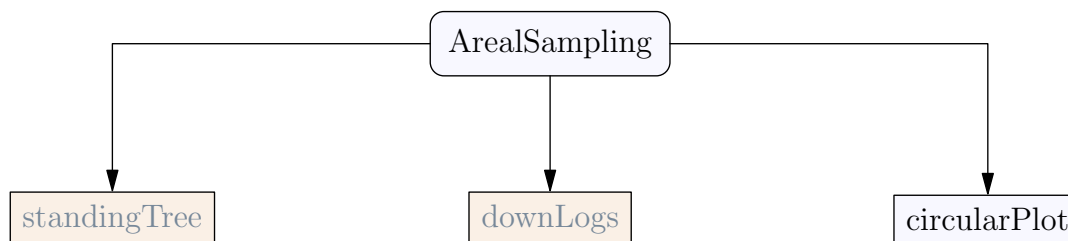


Figure 1: An overview of the “ArealSampling” class.

fig:AS

should be kept in mind that protocols within sampling methods, such as the “sausage” or “standup” protocols for down coarse woody debris (Gove and Van Deusen, 2011), are not “ArealSampling” methods per se. They could be defined as subclasses of the “circularPlot” class, but they really are characterized by their inclusion zones, and so we leave their definition for the “InclusionZone” class. In any case, if the divisions were to arise in further work, they would be defined as virtual classes, as they are in the “InclusionZone” class.

2 The “ArealSampling” Class

As mentioned above, this is the virtual base class, therefore, we really only care about its slots so we can see what will transfer to subclasses...

```
R> getClass('ArealSampling')
```

```
Virtual Class "ArealSampling" [package "sampSurf"]
```

```
Slots:
```

Name:	description	units	location	spID	spUnits
Class:	character	character	SpatialPoints	character	CRS

```
Known Subclasses: "circularPlot"
```

2.1 Class slots

- *description*: Some descriptive text about the object.
- *units*: A character string specifying the units of measure. Legal values are “English” and “metric.”
- *location*: This is a “SpatialPoints” representation of the location of the object. For example, in the “circularPlot” class, this is the fixed-radius plot center. It will often correspond to the `location` slot in the “Stem” object under sampling surface simulations. But there are exceptions: for example, under the ‘standup’ method, it will be at the large-end of the log, while under the ‘chainsaw’ method, it will be some point within the “sausage” shaped inclusion zone for protocol 1 in (Gove and Van Deusen, 2011).
- *spID*: A unique identifier that will be used in the eventual “SpatialPolygons” representation of the object.
- *spUnits*: A valid string of class “CRS” denoting the spatial units coordinate system (?CRS for more information) as in package `sp`.

3 The “circularPlot” Class

This is a subclass of “ArealSampling”, for fixed-area circular plots. It shares all the slots of the virtual class; in addition, it defines the following other slots...

```
R> showClass('circularPlot')
```

```
Class "circularPlot" [package "sampSurf"]
```

```
Slots:
```

Name:	radius	area	perimeter	description
Class:	numeric	numeric	SpatialPolygons	character

Name:	units	location	spID	spUnits
Class:	character	SpatialPoints	character	CRS

```
Extends: "ArealSampling"
```

3.1 “circularPlot” Class slots

The extra slots are defined as follows...

- *radius*: The fixed-plot radius in the correct units.
- *area*: The area of the plot in the correct units.
- *perimeter*: The “SpatialPolygons” object corresponding to the perimeter of the fixed-radius plot.

3.2 Object creation

One can use `new` to create a new object. However, as with other classes defined in `sampSurf`, the class is sufficiently tedious to create this way that a constructor function of the same name is provided. For example...

```
R> cp=circularPlot(37.237, units='English', center=c(x=10,y=3))
R> summary(cp)
```

```
Object of class: circularPlot
```

```
-----
fixed area circular plot
-----
```

```
ArealSampling...
```

```
  units of measurement: English
```

```
  spatial units: NA
```

```
  spatial ID: cp:20cc134c
```

```
  location...
```

```
    x coord: 10
```

```
    y coord: 3
```

```
    (Above coordinates are for plot center)
```

```
circularPlot...
```

```
  radius = 37.237 feet
```

```
  area = 4356.1141 square feet (0.1 acres)
```

```
  Number of perimeter points: 101 (closed polygon)
```

The arguments for the constructor are detailed in the help page. However, as an example, we see from the above `summary` output that the number of points defining the perimeter of the plot in the

“SpatialPolygons” object is given. It is in fact an argument to the constructor so the plot object can be created with as fine a perimeter of points as desired. The result will always be one more point than what is specified for the argument (default is 100 points), as it is necessary to close the polygon by repeating the starting point.

3.3 Plotting the object

The `plot` generic function has also been extended to be able to handle plotting of the objects of the “circularPlot” class. The arguments are again detailed in the help page, but here is a simple example...

```
R> plot(cp, axes=TRUE, showPlotCenter=TRUE, cex=2)
```

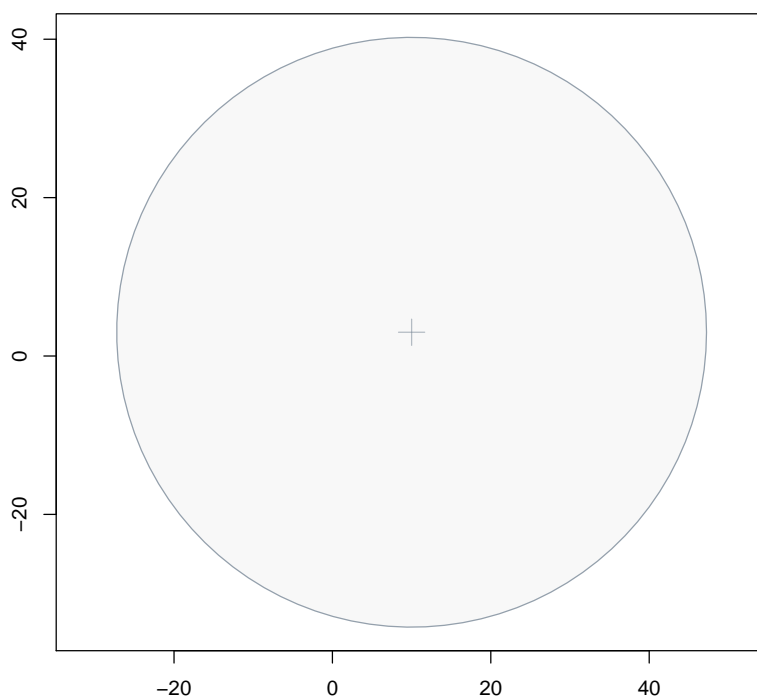


Figure 2: A “circularPlot” object.

fig:cp

In Figure ^{fig:cp}2, the `cex` argument specifies the size of the symbol for the plot center; other `par` arguments can also be included.

References

Deusen:2011

- J. H. Gove and P. C. Van Deusen. On fixed-area plot sampling for downed coarse woody debris. *Forestry*, 2011. Submitted August 2010. [1](#), [2](#), [3](#)