Tutorial 2 - Circular descriptives

Arjan Huizing

16 April 2018

Circular descriptives in R

To get started, install and load the package *circular* in R. This package contains functions for analyzing as well as plotting circular data.

```
# install.packages("circular")
library(circular)
```

For circular data there are two units of measurement that are commonly used, which are degrees and radians. A circle has a total of 360 degrees, or equivalently, a total of 2π (≈ 6.28 radians. Values cannot exceed this total, and if they do most functions will automatically scale them back to the circular range (e.g. 361 degrees equals 1 degree). To see this in action, we specify two small datasets, one in degrees and the other in radians. For this, we use the function **as.circular** and specify the measurement type with the argument units.

```
data.deg <- as.circular(c(0, 90, 210, 270, 360), units = "degrees")
data.rad <- as.circular(c(0, 90, 210, 270, 360), units = "radians")
```

The linear mean of the values we have specified is 186.

```
mean(c(0, 90, 210, 270, 360))
```

```
## [1] 186
```

Now let us calculate the mean direction in both circular datasets. The package *circular* contains the function **mean.circular**, which integrates with the base R function for calculating the mean. Lets give it a try.

```
mean(data.deg)
```

```
## Circular Data:
## Type = angles
## Units = degrees
## Template = none
## Modulo = asis
## Zero = 0
## Rotation = counter
## [1] -23.79398
```

```
mean(data.rad)
```

```
## Circular Data:
## Type = angles
## Units = radians
## Template = none
## Modulo = asis
## Zero = 0
## Rotation = counter
## [1] 1.400521
```

As we can see, the mean, mean direction of values in degrees and mean direction of values in radians are all very different. Where the regular mean simply takes the sum of all values and divides them by the amount of values, the mean direction takes account of the fact that data is on a circular continuum. In the dataset with values in degrees, the datapoints 90 and 270 are both the same distance from 0. Finally, the values of 0 and 360 are identical as well. The resultant mean direction is -23.8 degrees, which is the same as 336.2 degrees.

In our dataset with values in radians, all values except for 0 exceed 6.28. Thus when calculating the mean direction, all these values are rescaled. The mean direction is 1.4, which is about 80 degrees. In fact, the way we entered our data in radians suggests that we first wanted to convert a set of data in degrees to radians and then work with them as radians. To do that, we could have done;

```
data.rad <- as.circular(c(0, 90, 210, 270, 360) * pi / 180, units = "radians")
```

or even just

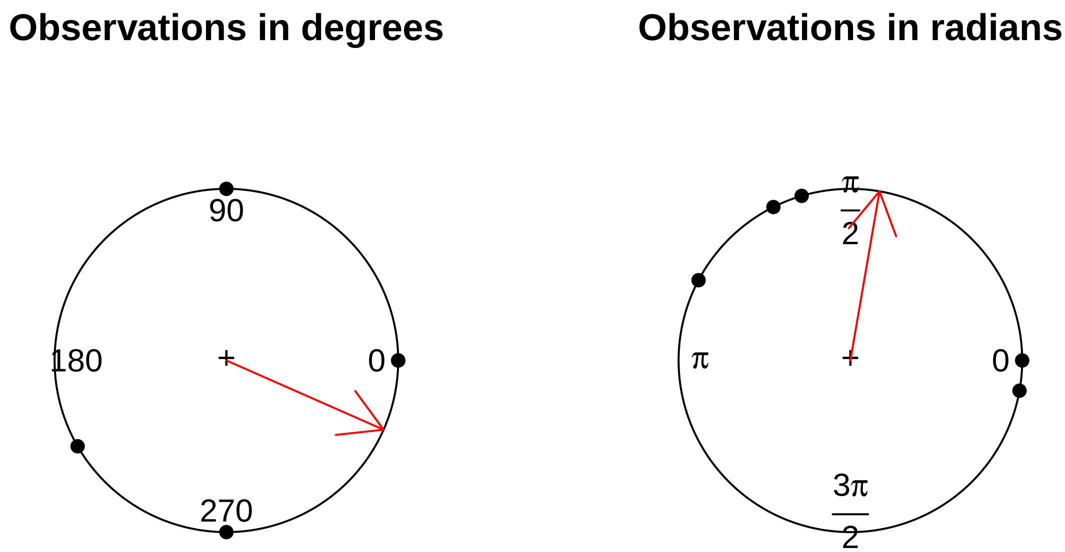
```
data.rad <- conversion.circular(data.deg, units = "radians")
```

This shows that we must be very careful: the data entered in `as.circular` must match the units that are given. For the sake of this tutorial, we will keep working with the original dataset, which interprets the values 0, 90, 210, 270 and 360 as radians directly.

Plotting

We can get a visual representation of both datasets using the **plot.circular** function. The function **arrows.circular** allow us to add arrows to the plot, which I have used to mark the mean directions in both.

```
par(mfrow = c(1,2))
plot(data.deg, main = "Observations in degrees")
arrows.circular(x = mean(data.deg),col = "red")
plot(data.rad, main = "Observations in radians")
arrows.circular(x = mean(data.rad),col = "red")
```



As you can see, we specified two very different datasets. Specifying the incorrect unit of measurement is a simple mistake one can make when analyzing circular data.

Other than a mean direction, we can also calculate a mean resultant length using **rho.circular**. This is a measure of *concentration* that can take on values between zero and one. One indicates that all values of the dataset are concentrated at a single location. A value close to zero indicates a higher spread. Interestingly enough, zero itself does not necessarily mean complete dispersion.

```
rho.circular(data.deg)
```

```
## [1] 0.2478627
```

```
rho.circular(data.rad)
```

```
## [1] 0.4352109
```

Our dataset in radians has a higher concentration of values than our dataset in degrees. This is congruent with the spread visible on the plots we made earlier.

Other than the mean direction and mean resultant length, we can also obtain a circular variance and a circular standard error. The variance is the inverse of the mean resultant length. The function **var.circular** will give you this value, but it is also easily obtained by $1 - \rho$. The standard deviation is *not* obtained using the square root of the variance, but rather using $(-2\log{(1-var)})^0.5$.

```
var(data.deg)
```

```
## [1] 0.7521373
```

```
sd(data.deg)
```

```
## [1] 1.670258
```