

# Tutorial 1 - An introduction to circular data

*Arjan Huizing*

*2 April 2018*

## What is circular data?

Most researchers will be familiar with the concept of measurement levels of variables. The most popular two being continuous data and categorical data. Circular data is another type of measurement with properties that distinguish it from other data types. The main property of circular data is the fact that it is periodical. What this means is that there is no true start or end point, but rather a continuous scale that loops back on itself. An example would be the time of day. As time passes the hands on the face of a clock will inevitably be bound to pass the same 12 hours over and over. This also means that distances can not be interpreted as linear. A difference in time between 1 and 3 is the same as the one between 1 and 11. Another quality of circular data is that every value can be seen to have an opposite value. When we analyze direction, we would say that north is the opposite of south, and east is the opposite of west. Circumplex models also use this quality to create a continuum of related values, where opposite values are 180 degrees apart. Examples of this from social and behavioural science are Schwartz' theory of basic values, Russel's mood scale and Leary's interpersonal circumplex.

Depending on the type of data we want to analyze, we use different methods. For example, if we want to predict a categorical outcome a binomial or multinomial logistic regression is preferred to using a linear regression. In the same vein, circular data should ideally be analyzed using methods tailored to the unique attributes of such data. Analyzing circular data as either categorical or continuous data may lead to issues of validity, as even something as simple as a mean value can result in very different outcomes if analyzed as continuous instead of circular as we will see.

## 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)
```

When using circular data there are two types of measurement that are commonly used. Degrees and radians. A circle has a total of 360 degrees, or a total of  $2\pi$  (6.28) radians. Values cannot exceed this total, and if they do most functions will automatically scale them back to the circular range (eg 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 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
```

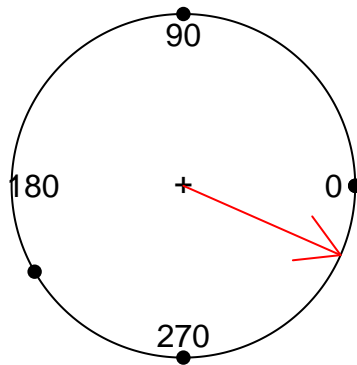
As we can see, the mean, mean direction of values in degrees and mean direction of values in radians are all very different. Where the regular mean simply takes the sum of all values and divides them by the amount of values, the mean direction takes account of the fact that data is on a circular continuum. 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, 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.

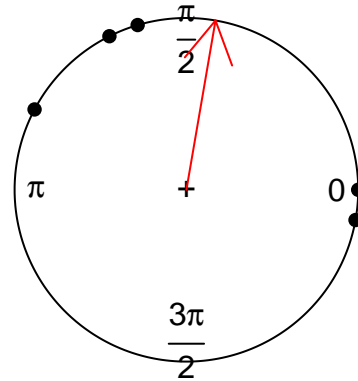
We can get a visual representation of both datasets using the **plot.circular** function. The function **arrows.circular** allow us to add arrows to the plot, which I have used to mark the mean directions in both.

```
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")
```

## Observations in degrees



## Observations in radians



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. Both values are based on the value for the mean resultant length. In the circular package, the variance is derived by taking the inverse of the mean resultant length and multiplying that value by 2. The angular deviation is the square root of the angular variance.

```
angular.variance(data.deg)
```

```
## [1] 1.504275
```

```
angular.deviation(data.deg)
```

```
## [1] 1.226489
```