# Von Mises Distribution Implementation in Python

Vedant Mehta, Kanchan Satpute

*Texas A&M University, College Station, Texas*

**Abstract:** This article tries to bridge the gap between the quality of package available for R and Python for circular statistics and von Mises distribution in specific. There is a function available in Python to generate random deviates from vonMises distribution. But there are no functions available to calculate the probability density, cumulative distribution, quantiles, etc.

## 1. Introduction

Directional statistics or circular statistics is a sub-discipline of statistics that deals with directions, axes and rotation. Think of it as a regular linear data converted into a circular data by giving it attributes like rotation, angle, etc. Circular statistics is a lot different than linear statistics. Firstly, there is no true zero. Namely, 0 and 360 degrees are equal. So, labeling a value as high or low is arbitrary. Due to these characteristics, method of analysis of this kind of data changes completely. The kind of data that has angles, or periodicity, or does not have a true zero can be labeled as directional data. Some of the examples include temporal periods (e.g. time of day, month, hour, week, etc.), compass directions, daily wind directions, ocean currents, etc.[5]

Calculation of mean, median and variance of a circular data is quite different from that in linear statistics. If given a data of angles, it cannot be simply averaged like it is done in linear statistics.

Method for mean calculation:

**Example:** Given angular data $(\alpha_1, \alpha_2, \dots, \alpha_n)$. Calculate the sine and cosine of all the angles.

Further, $X = \frac{\sum_{i=1}^{n} \cos \alpha_i}{N}$ and $Y = \frac{\sum_{i=1}^{n} \sin \alpha_i}{N}$. Also, $\bar{r} = \sqrt{X^2 + Y^2}$. So, mean cosine will be $\cos \bar{\alpha} = \frac{X}{r}$ and mean sine will be $\sin \bar{\alpha} = \frac{Y}{r}$. Finally, mean angle will be

$$\theta_r = \arctan\left(\frac{\sin \bar{\alpha}}{\cos \bar{\alpha}}\right)$$

Decide the resultant quadrant in following way: (Figure given for reference)

Sin +, cos + : mean angle computed directly

Sin +, cos - : mean angle = $180 - \theta_r$

Sin -, cos - : mean angle = $180 + \theta_r$

Sin -, cos +: mean angle = $360 - \theta_r$

Circular variance measures variation in the angles about the mean direction.
Variance $V = 1 - \bar{r}$. So it ranges from 0-1. When the variance is 1, it means the vectors are concentrated in one direction. Value of 0 means the vectors are equally dispersed around the circle.

There is another kind of data known as *bimodal data.* When data have opposite angles they are said to have diametrically bimodal circular distributions. The mean angle of diametrically bimodal data is orthogonal (at right angle) to the true mean. There is a procedure called *angle doubling* to deal with the diametrically bimodal data. But this article won't be discussing on that topic.

There are different types of distributions defined. Generally speaking, any kind of probability density function can be wrapped around the circumference of a circle. Von Mises distribution is one of the circular distributions that are defined in circular statistics and can be considered as analogous to normal distribution in linear statistics. Also, it is a close approximation to "wrapped normal" distribution.

Probability density function is given by:

$$f(x|\mu,\kappa) = \frac{1}{2\pi I_o(\kappa)} exp[\kappa\, cos(x-\mu)]$$

where, $I_o(\kappa)$ – Modified Bessel function of zero$^{th}$ order.

$\mu$ – measure of location (similar to mean in Normal distribution)

$\kappa$ – measure of concentration ( $1/\kappa$ is analogous to $\sigma^2$ )

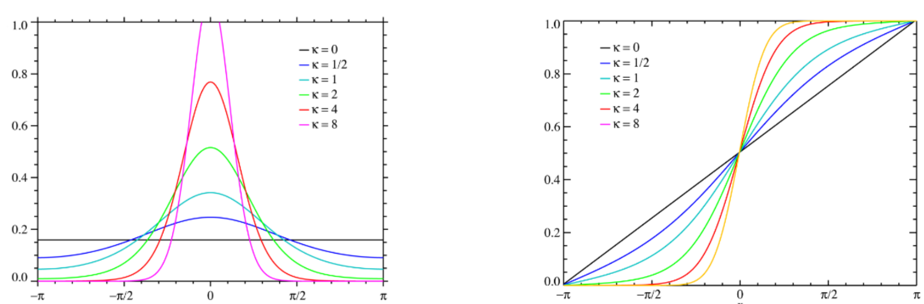
$I_o(\kappa)$ is defined as:

$$I_0(x) = \sum_{r=0}^{\infty} \frac{1}{r!\,r!}\left(\frac{x}{2}\right)^{2r}$$

$$= \frac{1}{2\pi}\int_0^{2\pi} e^{x\cos\theta}\, d\theta$$

The above equation is the zero$^{th}$ order modified Bessel function.

When $\kappa$ = 0, Von Mises distribution reduces to the uniform distribution. As $\kappa$ increases, the von Mises distribution approaches normal distribution.




Left panel depicts the PDF and right panel depicts the CDF of the von Mises distribution.

In the right panel, as $\kappa$ increases, the S curve will gradually become a straight line. That makes sense in a way, that as the concentration increases, the probability is more cumulated near the mean, i.e. zero in our case.

## 2. Methods

We have used the vonMises function in the circular package of R as a reference for generating the algorithms for each method.

**rvonmises(n, mu, kappa)**

Description – A method for generating random numbers for a von Mises circular distribution.

Arguments –

n – number of observations

Examples -

**dvonmises(x, mu, kappa)**

Description – A method for calculating the probability density at the given points for a von Mises circular distribution.

Arguments –

x – A vector containing the points at which the density is to be calculated. The object is from class 'circular'

log – logical; if True, probabilities p is given as log(p). The default value for log is given as False.

Examples –

**pvonmises(q, mu, kappa)**

Description – Method used to calculate the cumulative distribution at the given points for a von Mises distribution.

Arguments –

q – A vector containing the points at which the distribution is to be calculated. The object is from class 'circular'

tol – the precision in evaluating the distribution function. Default value = 1e-20

Examples –

**qvonmises(p, mu, kappa)**

Description – A method used to calculate the quantiles for the given probabilities for a von Mises distribution.

Arguments –

p – A vector containing the probabilities at points at which the quantiles are to be calculated. The object is from class ‘circular’

from_ - a value used for evaluating pvonmises and qvonmises. Default = None

tol – machine epsilon value raised to 0.6

Examples –

Common arguments for all the methods:

mu – The mean direction of the distribution. This object is from class ‘circular’

kappa – non-negative value for the concentration of the distribution

## 3. Results and Discussion

We run the functions pvonmises, qvonmises, dvonmises with various values of parameters mu and kappa. Below shown is the table that shows the comparison of the values obtained in R and values obtained by the package we built in Python.

| Method | R | Python |
|---|---|---|
| pvonmises(2, 1, 6)<br>pvonmises([2, 0.8], 2, 6) | [0.9888944]<br>[0.5 , 0.003595458] | [0.988894]<br>[0.5 , 0.00359546] |
| dvonmises(0.5, 1, 6)<br>dvonmises([1, 3], 3, 6) | [0.4581463]<br>[1.949157e-04, 9.54982e-01] | [0.45814625]<br>[1.949157e-04, 9.54982e-01] |
| qvonmises(0.5, 1, 6)<br>qvonmises([0.2, 0.6], 2, 7) | [1]<br>[1.67413597, 2.09767203] | [1]<br>[1.67413597, 2.09767203] |

Now, we will plot some graphs to demonstrate how precise our values are when compared to those in R

When we run the function rvonmises(n=1000, mu=1, kappa=1), it generates following output in R and Python respectively.

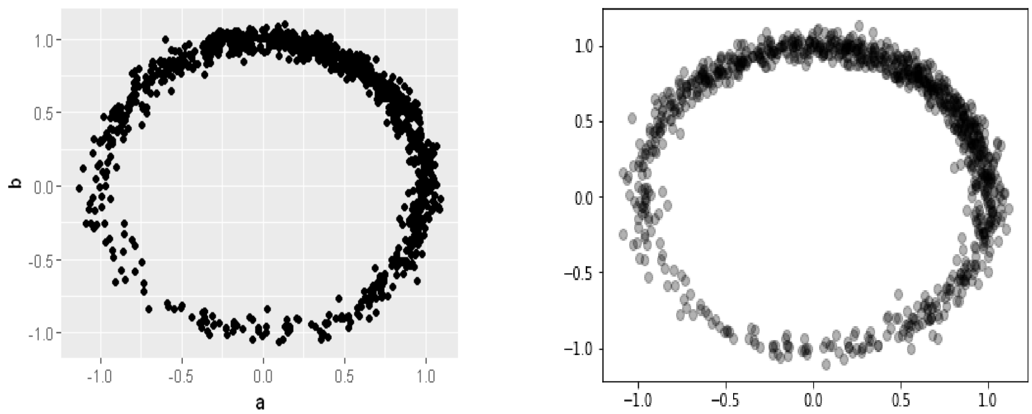



*Figure 1: rvonmises in R (left panel) and Python (right panel)*

When we run the function dvonmises(x = np.linspace(-pi, pi, 1000), mu=1, kappa=6), it generates following output in R and Python respectively.

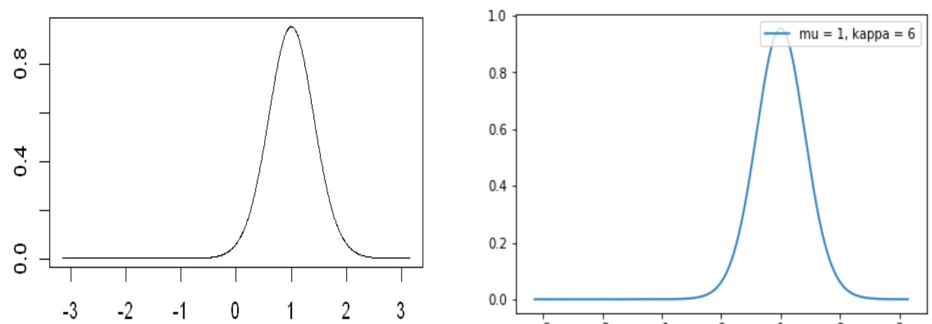



*Figure 2: dvonmises in R (left panel) and Python (right panel)*

## 4. Future Scope

We need to make the package more robust so that the function can accept different kind of inputs. When we ran the benchmarking tests, we saw that our code took longer time to execute as compared to that in R. So we need to optimize the code in order to decrease the execution time. We can include other functions from the 'circular' package of R into Python.

## 5. Reference


[1] https://www.researchgate.net/figure/Wind-data-for-KRDM-the-nearest-FAA-weather-reporting-station-at-the-Redomond-OR_fig5_261417337

[2] https://ncss-wpengine.netdna-ssl.com/wp-content/uploads/2013/01/Rose-Plot.png

[3] http://webspace.ship.edu/pgmarr/geo441/lectures/lec%2016%20-%20directional%20statistics.pdf

[4] https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Circular_Data_Analysis.pdf

[5] https://en.wikipedia.org/wiki/Von_Mises_distribution

[6] https://packaging.python.org/tutorials/distributing-packages/#your-package

[7] https://r-forge.r-project.org/scm/viewvc.php/pkg/R/vonmises.R?view=markup&root=circular

[8] https://cran.r-project.org/web/packages/circular/circular.pdf