Applied Mathematics and Computational Intelligence
Volume 10, No.1, Dec 2021 [278-287]

Graphical Summaries of Circular Data with Outliers Using Python Programming Language

Nur Syahirah Zulkipli [1*], Siti Zanariah Satari [1], Wan Nur Syahidah Wan Yusoff[1]

[1] Centre for Mathematical Sciences, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Lebuhraya Tun Razak, 26300 Gambang, Kuantan, Pahang, Malaysia.

[*] Corresponding author: syahirahzulkipliwork@gmail.com

Received: 10 August 2021; Accepted: 10 November 2021; Available online: 29 November 2021

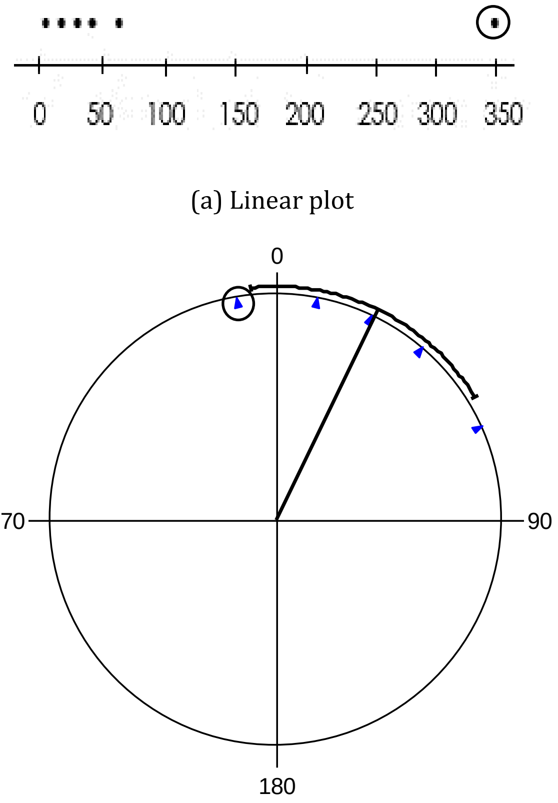
ABSTRACT

Graph in statistics is used to summarise and visualise the data in pictorial form. Graphical summary enables us to visualise the data in a more simple and meaningful way so that the interpretation will be easier to understand. The graphical summaries of circular data with outliers is discussed in this study. Most of the time, people use linear data in real life applications. Other than linear data, there is another data type that has a direction which refers to circular data and it is different from linear data in many aspects such as in descriptive statistics and statistical modeling. Unfortunately, the availability of statistical software specialises in analysing circular data is very limited. In this study, the graphical summaries of circular data are plotted using the in-demand programming language, Python. The Python code for generating graphical summaries of circular data such as circular dot plot and rose diagram is proposed. The historical circular data is used to illustrate the graphical summaries with the existence of outliers. This study will be helpful for those who are started exploring circular data and choose Python as an analysis tool.

Keywords: Circular Data, Circular Dot Plot, Rose Diagram, Outlier, Python

1 INTRODUCTION

Visual display such as graph in statistics is used to summarise and visualise data in pictorial form. It is important to present the data or analysis in graphical summary before starting any further statistical analysis. Graphical summary enables us to visualise the data in a more simple and meaningful way so that the interpretation will be easier to understand. Most of the time, people use linear data in real life applications. Data such as daily expenses, sales profit and internet usage are recorded daily and these kinds of data are known as linear data. Other than linear data, there is another data type that has a direction which refers to circular data [1]. Circular data is widely found in the field of meteorology and biology where researchers are interested to investigate the direction of wind and animals. Circular data gain its popularity since early 1970 and researchers from applied science such as [1 - 5] are working hard to make a contribution in the field of circular data. This kind of data have many unique and novel characteristics, both in terms of modeling and their statistical procedures [1]. The different between linear data and circular data can be illustrated by plotting the following dataset of 10°, 25°, 40°, 65° and 350° in a unit circle (degree) using linear and circular approaches as given in Figure 1.



(a) Linear plot

(b) Circular plot

Figure 1 : Illustration of univariate circular data using linear and circular plots

Figure 1(a) is a linear plot when the dataset is analyse using linear approach. Meanwhile, Figure 1(b) is a circular plot when the circular approach is used to analyse the dataset. Figure 1(a) shows that if this dataset is illustrated using linear approach, the observation 350° which highlighted in the black circle is considered as outlier since this observation deviates far away from the others. However, when this dataset is considered as a sample of circular data, the observation 350° is consistent with the others as shown in Figure 1(b). Therefore, this illustration gives a clear picture of the different between linear data and circular data especially in the application of detecting an outlier. Observations in a dataset which deviates far from other observations are defined as outliers. According to [3], outlier is referred to the object or observation that is not consistent with the other observations in a dataset and the existence of outliers in circular data can give a large effect to the parameter estimates and inferences.

Till date, researchers still working hard to develop statistical procedures and developing more statistical software that specialised for circular data. According to [6], the user-friendly statistical software specialised for circular data is limited. Currently, there are few statistical software and high-level programming languages available to analyse circular data such as ORIANA, Matlab, R and Python. Python and R are open source programming languages and both are commonly used by data analysts and statisticians. According to [7], R programming language is frequently used by the researchers to analyse circular data since the package for circular statistics called 'circular' and 'CircStats' has been released in 2017 and 2018, respectively. However, [8] stated that, the usage of Python language has been soaring since the early 2000s in industrial applications and research, while R still a popular language for traditional data analytical procedures. Recently, Python rank to be the top followed by Java, C, C++, JavaScript and R [9].

Python has been developed by Guido Van Rossum and first released on February, 1991. Python is widely known as high-level programming language and very useful for general-purpose programming. Nowadays, Python is the most commonly used for data science programming language asides R and according to [10], R is more complex to learn compared to Python. According to [11], Python is a programming language that can be easily to learn and understood by the user especially students. Python has simple syntax but it is strong integration programming language, which can be the reasons why Python has been widely used by many scientists and developers [11]. Furthermore, since Python is already known as the integrated programming language, it has many integrated development environments or commonly known as IDEs such as Jupyter, PyCharm, Spyder, IDLE and etc. Over the years, Python has rapidly developed huge libraries for data science such as Numpy, Pandas, Scipy, Matplotlib, StatsModel and Seaborn. These Python libraries are very familiar among data analyst and statisticians. Unfortunately, the library that specialised for circular statistics such as 'pycircstat' and 'spicy.stats' are still not fully developed for a certain circular data analysis. Here, we notice that the existing packages for circular data are very limited to generate a graphical summary for circular data.

Thus, the main objective of this study is to develop Python coding for graphical summaries of univariate circular data such as circular dot plot and rose diagram with available Python libraries. At the end of this study, the Python code for circular dot plot and rose diagram will be proposed. This study will be helpful for those who are just started exploring circular data and choose Python as analysis tool. In the next Methodology section, the characteristics of circular dot plot and rose diagram will be discussed. The historical circular data will be used in this study to illustrate the graphical summaries of univariate circular data. Next, the Python code will be displayed in the 'Proposed Python Code' section and result of this study will be discussed.

2 MATERIAL AND METHODS

There are many graphical summaries that have been developed for circular data and due to the bounded property of circular data, the graphical summaries for linear data is not appropriate to be used for circular data. Graphical summaries of circular data such as circular dot plot, circular histogram and rose diagram are commonly used in exploratory circular data analysis. The circular dot plot is commonly used to visualise the raw circular data by plotting each observation as a point on the circle circumference. Circular dot plot very useful to visualise the spread of the data and describe the distribution of the data. Another simple graphical summary that commonly used is circular histogram. Each bar in a circular histogram is centered as the midpoint of corresponding group angles which are similar to histogram for linear cases. The visual impression given by circular histogram may be sensitive to the grouping used, as it analogous to histograms on the real line.

Somehow, circular histogram can be useful to transform into linear histogram by cutting the circular histogram at a suitably chosen point on the circle. Note that, to transform circular histogram to linear histogram, an interval of width 360° can be used.

Rose diagram or also called rose plot is another alternative of the circular histogram where the bars of the circular histogram are replaced by sectors. The area of each sector must be proportional to the frequency of the corresponding group. The circular histogram and rose diagram are very useful to visualise the circular data in graphical representation in order to determine the distribution of the dataset. The general forms of circular distributions can be uniform, unimodal, bimodal and multimodal distributions. A circular data that shows a symmetrical, skewed or peak form of distribution are refer as a unimodal distribution. In this study we will only propose the graphical summaries such as circular dot plot and rose diagram for univariate circular data using Python. There are many libraries for data visualisation available in Python such as Matplotlib, Seaborn and Plotly. The libraries that will be used in this study are Numpy and Matplotlib. According to [12], Matplotlib is a powerful graphics library for data visualisation in Python which widely used and it is commonly work together with Numpy, Pandas and other relevant libraries. The documentation of these libraries can be referred from [13-15]. The Python programming language version 3.8 and Python IDE, Jupyter Notebook will be used in this study.

3 DATA APPLICATION

In this study, a historical dataset is used to illustrate the application of graphical analysis for circular data. The data involves is the direction of northern cricket frogs, Acris Crepitans data defined by [16]. A data of homing ability of the 14 northern cricket frogs, Acris Crepitans was recorded in a series experiment by [16]. Homing ability is the inherent animal's ability to navigate towards the original location either a home territory or a breeding spot through the unfamiliar areas. The frogs were collected from the mudflats of abandoned stream meander near Indianola, Mississippi. After 30 hours of enclosure within a dark environmental chamber, the frogs were released and the directions taken by the frogs were recorded. The Frog data are given in Table 1.

Table 1 : A dataset of 14 directions of northern cricket frogs, *Acris Crepitans*.

Observation, $\boldsymbol{\theta_i}$	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_2}$	$\boldsymbol{\theta_3}$	$\boldsymbol{\theta_4}$	$\boldsymbol{\theta_5}$	$\boldsymbol{\theta_6}$	$\boldsymbol{\theta_7}$	$\boldsymbol{\theta_8}$	$\boldsymbol{\theta_9}$	$\boldsymbol{\theta_{10}}$	$\boldsymbol{\theta_{11}}$	$\boldsymbol{\theta_{12}}$	$\boldsymbol{\theta_{13}}$	$\boldsymbol{\theta_{14}}$
Frog's direction (in degrees)	104	110	117	121	127	130	136	145	152	178	184	192	200	316

The Frog data given in Table 1 is widely used to illustrate the existence of outlier in univariate circular data. In literature, Observation 14 has been detected as outlier such as in [1], [17], [18] and [19]. Therefore, the Frog data will be visualised in circular dot plot and rose diagram to confirm the outlier by using proposed Python code and it will be discussed in the next section.

4 PROPOSED PYTHON CODE

In this section, the Python code for circular dot plot and rose diagram are proposed and summarised in Table 2 with the corresponding Python libraries for each graphical plot. A user-friendly IDE, Jupyter Notebook is used in this study to write and run the Python code. Besides Jupyter Notebook, Python IDE such Spyder which has been used in [20] also recommended. All listed libraries in Table 2 must be imported into Python before execute the code. The Python code in Table 2 are the functions of circular dot plot and rose diagram. `def` is the keyword used in Python to define a function and the function name is followed by the parameter(s) in the bracket `()`. The circular dot plot and rose diagram can be visualised by calling the function `circplot(data)`and `roseplot(data1,data2)`. Hence, by calling these functions means that the circular dot plot and rose diagram can be executed either directly or through a nested function.

Table 2 : Proposed python code for graphical summaries of circular data.

<table>
<tr><th>Graphical summary</th><th>Python code</th><th>Library</th></tr>
<tr><td>Circular dot plot</td><td><pre>
import numpy as np
import matplotlib.pyplot as plot

# Define the circular dot plot function

def circplot(data):
    plot.axes(projection='polar')
    data = data
    for radian in data:
        plot.polar(radian, 5, '.', color='blue')
    plot.show()
</pre></td><td>Numpy
Matplotlib.pyplot</td></tr>
<tr><td>Rose diagram</td><td><pre>
import numpy as np
import matplotlib.pyplot as plt

# Define the rose plot function

def rose_plot(ax, angles, bins=16, density=None,
              offset=0, label_unit="deg",
              start_zero=False, **param_dict):
    angles = (angles+np.pi)%(2*np.pi)-np.pi

    if start_zero:
        if bins % 2:
            bins += 1
        bins = np.linspace(-np.pi, np.pi,
                           num=bins+1)

    count, bin = np.histogram(angles, bins=bins)
    widths = np.diff(bin)
</pre></td><td>Numpy
Matplotlib.pyplot</td></tr>
</table>

```
    if density is None or density is True:
        area = count / angles.size
        radius = (area / np.pi)**0.5
    else:
        radius = count

    ax.bar(bin[:-1], radius, zorder=1,
           align='edge', width=widths,
           edgecolor='C0', fill=False,
           linewidth=1)
    ax.set_theta_offset(offset)
    ax.set_yticks([])

    if label_unit == "rad":
        label = ['$0$', r'$\pi/4$',
                 r'$\pi/2$', r'$3\pi/4$',
                  r'$\pi$', r'$5\pi/4$',
                 r'$3\pi/2$', r'$7\pi/4$']
        ax.set_xticklabels(label)

# Define the function to visualise rose diagram

def roseplot(data1,data2):
    fig, ax = plt.subplots(2, subplot_kw =
dict(projection='polar'), figsize=(20,8))
    # to plot in degrees unit (by default)
    rose_plot(ax[0], data1)
    # to plot in radians unit
    rose_plot(ax[1], data2,
              label_unit="rad")
    fig.tight_layout()
```

Table 3 shows the Python code to visualise the circular dot plot and rose diagram by using Frog data. The Frog dataset is already saved in *csv* file and the Python code to import the dataset is shown in Table 3. Pandas library is used to read or load the dataset. Therefore, `read_csv()` function from Pandas library is used to execute the code. Note that, the data file must be located at the same path of the Python code. In addition, the unit of circular data must be in radian and hence, the circular data need to be converted from degrees unit to radians unit by using Numpy library. After importing the Frog dataset, the graphical summaries for Frog data can be executed by calling the functions in Table 2 and the example of code execution of the circular dot plot and rose diagram funtions are shown in Table 3.

Table 3 : Python code to visualise graphical summaries of Frog data.

Action	Python code	Library

Import data	`import pandas as pd`
	`frogdeg = pd.read_csv('Frogdeg.csv')`
	`frog_rad = np.deg2rad(frogdeg.fdirect)`
	`# Frogdeg.csv is the file name that saved by the author`
	`# fdirect is the column name of frog dataset in Frogdeg.csv file`
		Pandas
		Numpy

Visualised Frog data	`circplot(frog_rad)`
	`roseplot(frog_rad, frog_rad)`
		Numpy
		Matplotlib.pyplot

5 RESULT AND DISCUSSION

The result and finding of the proposed objective are given here. The directions of Frog data are visualised as shown in Figure 2 and Figure 3, respectively.

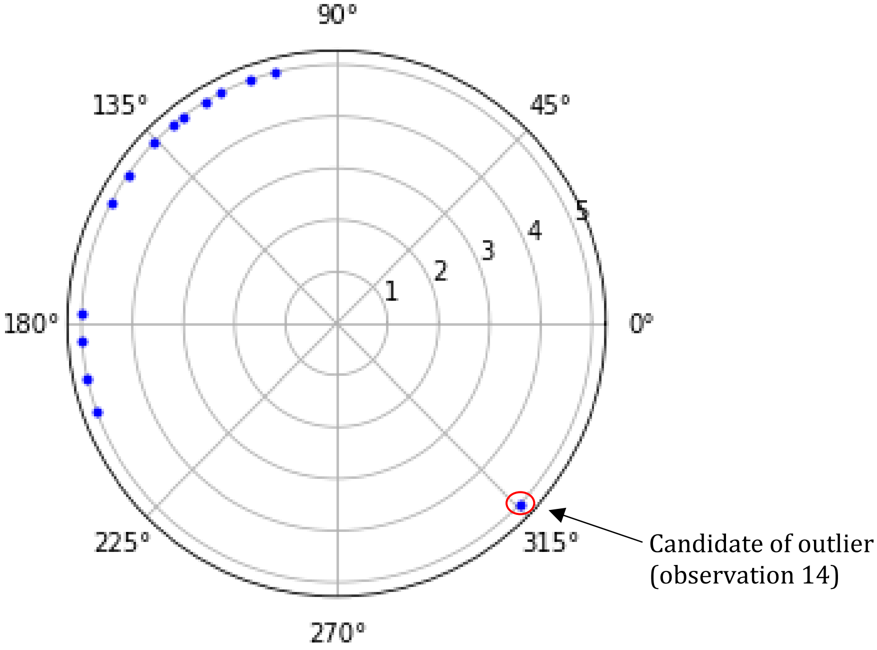


Figure 2 : Circular plot of directions of 14 Northern Cricket Frogs.

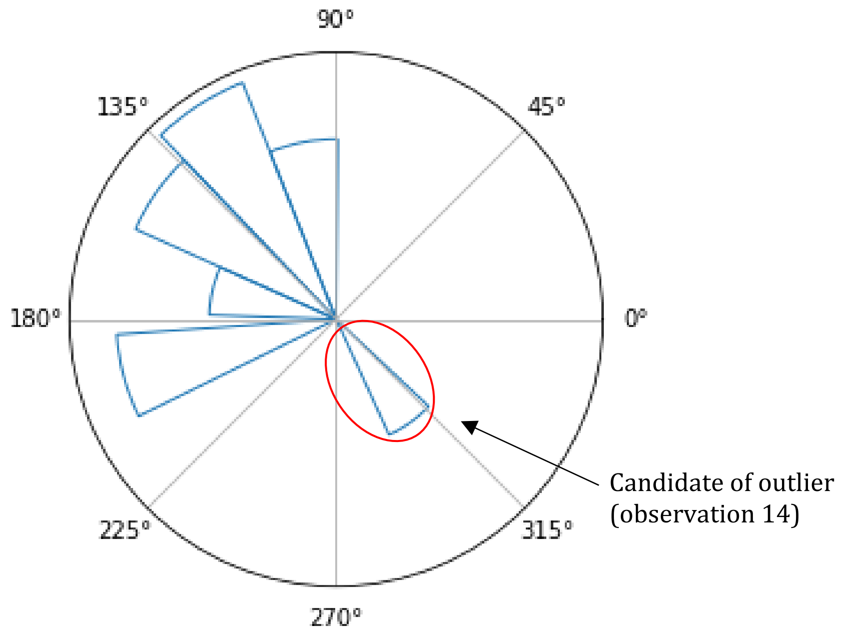


Figure 3 : Rose diagram of directions of 14 Northern Cricket Frogs.

From Figure 2 and Figure 3, observation 14 is suspected to be an outlier since it is located far away from the rest of other observations. This observation is located at the fourth quadrant which is 5.5152 radians or 316°. Moreover, this observation may affect the distribution of data and it is important to investigate further on this observation. On the other hands, the other 13 observations are found to be concentrated towards the center of dataset. Hence, the proposed Python code is applicable to visualise the graphical summaries of Frog data and confirms the finding made by other researchers in [1], [17], [18] and [19].

6 CONCLUSION

In summary, this study found that Python is applicable to be used for generating the graphical summaries for circular data especially using circular dot plot and rose diagram. Generally, Python libraries such as Numpy and Matplotlib.pyplot are used to create the circular dot plot and rose diagram functions for circular data. Besides that, Pandas library also used to import or load the dataset. The steps for execute or run the Python code for circular dot plot and rose diagram are explained clearly in this study. It is also found that, circular dot plot and rose diagram are applicable to be used for the historical circular data, Frog data. The graphical summaries also be able to identify possible outlier in Frog data which result similar with previous studies such as [1], [17], [18] and [19] where these studies used Splus and R programming language to visualise the graphical summaries.

In conclusion, the graphical summaries for circular data can be plotted using the proposed code since the library specialised in data visualisation is available in Python. Thus, in future, the proposed code of circular dot plot and rose diagram is suggested to be used in other data visualisation project such as in graphical user interface (GUI) or in any web application since Python can build these projects. In addition, Python is very simple to be learn especially for those who are new to programming and it is recommended to start with Python and use Python IDE such as Jupyter Notebook since it is easy to be used and also supported other programming languages. Lastly, this study is beneficial for those who are started exploring circular data and decided to use Python as their circular analysis tool.

ACKNOWLEDGEMENT

Authors would like to thank all the associate editors and referees for their thorough reading and valuable suggestions which led to the improvement of this paper. The Ministry of Higher Education Malaysia and Universiti Malaysia Pahang is acknowledged for the financial support received for this study. (FRGS/1/2019/STG06/UMP/02/6 and UMP Internal Grant: RDU190363 and RDU1901168).

REFERENCES

[1] S. R. Jammalamadaka and A. Sengupta, *Topics in Circular Statistics*. Singapore: World Scientific Publishing, 2001.

[2] K. V. Mardia, “Statistics of directional data,” Journal of the Royal Statistical Society: Series B (Methodological), vol. 37, no. 3, pp. 349-393, 1975.

[3] D. Collett, “Outliers in Circular Data,” Journal of the Royal Statistical Society, vol. 29, no. 1, pp. 50–57, 1980.

[4] D. J. Best and N. Fisher, “The bias of the maximum likelihood estimators of the von Mises-Fisher concentration parameters,” Communication in Statistics- Simulation and Computation, vol. 10, no. 5, pp. 493–502, 1981.

[5] N. I. Fisher, *Statistical Analysis in Circular Data*. New York, USA: Cambridge University Press, 1993.

[6] S. F. Hassan, A. G. Hussin, and Y. Z. Zubairi, “Analysis of Malaysian wind direction data using ORIANA,” Modern Applied Science, vol. 3, no. 3, pp. 115–119, 2009.

[7] N. S. Zulkipli, S. Z. Satari, and W. N. S. Wan Yusoff, “Descriptive analysis of circular data with outliers using Python programming language, ” Data Analytics and Applied Mathematics (DAAM), vol. 1 no. 1, pp. 31–36, 2020.

[8] C. L. P. Chen and C. Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on Big Data,” Information Sciences, vol. 275, pp. 314–347, 2014.

[9] S. Cass, “Top programming languages 2020,” *IEEE Spectrum*, July 22, 2021. [Online]. Available: https://spectrum.ieee.org/at-work/tech-careers/top-programming-language-2020.

[10] R. Sharma, “Top 6 data science programming languages 2021 [hand-picked],” *upGrad blog*, Jan. 8, 2021. [Online]. Available: https://www.upgrad.com/blog/data-science-programming-languages/.

[11] S. Marikala, “Python and its libraries in data science and related fields,” Data Science and Engineering, Vol. 1, no. 1, 2020.

[12] A. Hafeez and A. H. Sial, “Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python,” International Journal of Advanced Trends in Computer Science and Engineering, vol. 10, no. 1, pp. 277–281, 2021.

[13] “NumPy Documentation,” NumPy. Mar. 31, 2021. [Online]. Available: https://numpy.org/doc/

[14] “Matplotlib: Visualization with Python,” Matplotlib. Mar. 31, 2021. [Online]. Available: https://matplotlib.org/

[15] “Pandas Documentation,” Pandas. Mar. 31, 2021. [Online]. Available: https://pandas.pydata.org/docs/

[16] D. E. Ferguson, H. F. Landreth, and J.P. Mckeown, “Sun compass orientation of the northern cricket frog, Acris crepitans,” Animal Behaviour, vol. 15, no. 1, pp. 45-53, 1967.

[17] A. H. Abuzaid, A. G. Hussin, A. Rambli, and I. Mohamed, “Statistics for a new test of discordance in circular data,” Communications in Statistics—Simulation and Computation, vol. 41, pp. 1882–1890, 2012.

[18] E. A. Mahmood, S. Rana, H. Midi, and A. G. Hussin, “Detection of outliers in univariate circular data using robust circular distance,” Journal of Modern Applied Statistical Methods, vol. 16, no. 2, pp. 418–438, 2017.

[19] F. N. Badarisam, A. Rambli, and M. I. Sidik, “A comparison on two discordancy tests to detect outlier in von mises (VM) sample,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 9, no. 1, pp. 155–162, 2020.

[20] S. C. R. Nandipati, X. Chew, and K. W. Khaw, “Polycystic Ovarian Syndrome (PCOS) classification and feature selection by machine learning techniques,” Applied Mathematics and Computational Intelligence, vol. 9, pp. 65–74, 2020.