RESEARCH ARTICLE

INTERNATIONAL JOURNAL OF KINANTHROPOMETRY

ISAK

INTERNATIONAL JOURNAL OF KINANTHROPOMETRY

DOI: 10.34256/ijk2417

Using R Programming for Somatoploting

Dattaniranjan Nandikolmath [1,] *, Rupesh Samanchi [2], Barsha Rana [3], Aruna Hallikeri [4]

[1] Department of Studies in Anthropology, Karnatak University, Dharwad, Karnataka-580003, India.
[2] Department of Physiology, All India Institute of Medical Sciences, Guwahati-781101, India
[3] Grant Thornton, Bengaluru, Karnataka- 560093, India
[4] Department of Anthropology, Karnatak Arts College, Dharwad, Karnataka-580001, India
* Corresponding authors email: mathnandikol@gmail.com

DOI: https://doi.org/10.34256/ijk2417
Received: 17-02-2024; Revised: 04-04-2024; Accepted: 10-04-2024; Published: 30-04-2023

Resumen

Introducción: La antropometría es una técnica empleada para evaluar las dimensiones y proporciones corporales mediante el examen de la longitud, el ancho, la circunferencia y el espesor de los pliegues cutáneos. Es rentable, sencillo y fácilmente transportable, y puede utilizarse en diversas industrias. La somatotipificación es un método principal utilizado para clasificar el físico humano en función de tres componentes principales: endomorfia, mesomorfia y ectomorfia. Heath y Carter establecieron y modificaron los estándares de somatotipo, que continúan utilizándose para mediciones globales. Se han creado varias herramientas de software para el somatoploteo, incluidos SAS/GRAPH, Houcine, Orhan y modelos de aprendizaje automático. Sin embargo, la mayoría de estas herramientas no son de código abierto, lo que resulta en una laboriosa enumeración manual y dificulta la representación precisa de las diferencias entre grupos. Se requiere una herramienta precisa, funcional y de código abierto para categorizar somatotipos de tamaños de muestra extensos e ilustrar sus diferencias. **Métodos:** La programación R es un lenguaje de programación potente y versátil, particularmente popular en computación estadística y gráficos. Se utiliza ampliamente en diversos campos, como la bioestadística, la bioinformática y el análisis de mercados financieros. R incorpora conceptos de programación originales como la programación orientada a objetos, que los usuarios pueden utilizar de forma transparente. Este artículo presenta cómo utilizar la programación R como herramienta para el somatoploteo, presenta el código para el somatoploteo, inserta datos xey y ejecuta el programa para obtener un somatográfico. Utiliza datos antropométricos de 34 estudiantes escolares recopilados en Shindikurbet, Karnataka, de entre 10 y 12 años, recopilados a través de las pautas del protocolo ISAK para desarrollar somatotipos y trazarlos en mayor detalle. **Resultados:** el artículo demuestra de manera integral el uso de la programación R para trazar somatotipos en un Somatochart 2-D. Mediante este proceso, el lector puede desarrollar somatocartas de alta calidad en formatos de imagen o PDF. **Conclusión:** Este estudio explora el uso de la programación R, un software de código abierto, para el trazado de somatogramas y la generación de somatográficos. Este método ayuda a comprender información compleja, explicar hechos y guiar acciones en diversos campos. Ofrece procesamiento, análisis y presentación de datos accesibles, lo que facilita la implementación y el ahorro de presupuestos para estudiantes, investigadores e instituciones. Se podrían realizar más investigaciones para hacer que el código sea más fácil de usar en hojas de Excel o aplicaciones móviles.

Palabras Clave: Programación R, RStudio, Somatotipado, Somatochart, Somatotipos.

Abstract

Introduction: Anthropometry is a technique employed to evaluate body dimensions and ratios by examining body length, width, circumference, and skinfold thickness. It is cost-effective, uncomplicated, and easily transportable, and it may be used in diverse industries. Somatotyping is a primary method used to classify the human physique based on three main components: endomorphy, mesomorphy, and ectomorphy. Heath and Carter established and modified the standards of somatotype, which continue to be employed for global measurements. Several software tools have been created for somatoploting, including SAS/GRAPH, Houcine, Orhan, and machine-learning models. Nevertheless, most of these tools are not open-source, resulting in laborious manual enumeration and hindering the accurate representation of differences among groups. A functional, open-source, precise tool is required to categorise somatotypes of extensive sample sizes and illustrate their differences. **Method:** R programming is a powerful and versatile programme language, particularly popular in statistical computing and graphics. It is widely used in various fields, like biostatistics, bioinformatics, and financial market analysis. R incorporates original programming concepts like object-oriented programming, which users can use transparently. This paper introduces how to use R programming as a tool for somatoploting, introduces the code for somatoploting, inserts x and y data, and executes the program to get a somatochart. It uses anthropometric data of 34 school-going students collected in Shindikurbet, Karnataka, aged between 10 and 12 years, collected through ISAK protocol guidelines to develop somatotypes and further plot them. **Result:** The paper holistically demonstrates using R programming to plot somatotypes in a 2-D Somatochart. Using this process, the reader can develop high-quality somatocharts in image or PDF formats. **Conclusion:** This study explores using R programming, an open-source software, for somatoploting and generating somatocharts. This method aids in understanding complex information, fact-explaining, and guiding action in various fields. It offers accessible data processing, analysis, and presentation, making implementing and saving budgets for students, researchers, and institutions easy. Further research could be conducted to make the code easier to use in Excel sheets or mobile applications.

Keywords: R programming, RStudio, Somatotyping, Somatochart, Somatotypes

Introducción

Anthropometry offers a relatively inexpensive, simple, portable, and effective technique to assess body size and proportions by inspecting body length, width, circumference (C), and skinfold thickness (SF) (Wang et al., 2000). Its knowledge has been used in various fields, including nutrition, kinanthropometry, sports science, physiology, apparel designing, auxology, and forensic science. One of the prime methods in anthropometry is somatotyping, which classifies the human physique based on three primary components, i.e., endomorphy, mesomorphy, and ectomorphy. (Singh & Mehta, 2009, p 126). Sheldon (1954) defines somatotype as

"Somatotypes are morphophenotypic ranges along continua of variation which possess constantly recognisable characteristics and are the functional end products of the whole genetic and developmental complex."

Heath and Carter (1967) devised and revised the norms of somatotype, which are, to date, referred to and used for measurements worldwide. One of the essential and final steps in somatotyping is plotting a somatochart. A 2D-Somatochart is a pictorial representation of an individual's physique in a reuleaux triangle (Figure 1) (Siders & Rue, 1992) based on the three essential components, i.e., mesomorphy, endomorphy, and ectomorphy.

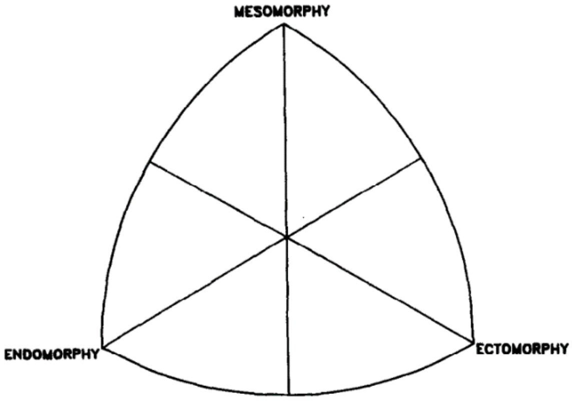


Figure 1. A 2-D Somatochart

A somatotype's x and y coordinates are calculated using endomorphy, mesomorphy, and ectomorphy values. While the x value is ‘ectomorphy – endomorphy,’ the y is calculated using the formula ‘2 x mesomorphy - (endomorphy + ectomorphy)’ (Tóth et al., 2014).

A range of software tools have been developed for somatotyping. Satake (1993) introduced a program for plotting somatotypes using SAS/GRAPH, providing a convenient method for visualising somatotype distributions. Houcine (2014) designed software for counting body composition and somatotype, facilitating the calculation of these parameters for many individuals. Orhan (2013) used the Somatotype 1.1 program for somatotype calculations and analyses, comparing the somatotype values of football players in different positions. Chiu et al. (2022) developed machine-learning models for estimating somatotypes from 3D body scanning data, demonstrating the potential for automated somatotyping. These studies highlight the diverse range of software tools available for somatotyping, each with unique features and applications.

However, most of these software tools are not open-source but rather are to be purchased. Hence, in most academic setups, students and researchers plot the somatotypes manually, i.e., calculating x and y values and drawing lines on paper while tracking the coordinates. Manual enumeration of the somatotype is handy when we have less number of subjects to reproduce on the somatoplot. Nevertheless, it becomes tedious when the numbers are in hundreds of subjects. Representing group variation is challenging when the sample size is more extensive. Further, there is a compromise in plotting accuracy as decimal variations are neglected. These issues specify that there is a need for a tool that is workable without much complication, open-source, accurate, and can enumerate somatotypes of large sample sizes and represent variations among them.

About R programming

R is a powerful and versatile programming language, particularly popular in statistical computing and graphics (Tuimala & Kallio, 2013). It is a language developed to implement further S language, which was developed in 1995 by Ross Ihaka and Robert Gentleman as part of the GNU project, released in 1997 (Anand, 2022). It is widely used in biostatistics, bioinformatics, and financial market analysis (Horgan, 2012). The language is known for its expressive and compact nature, with a wide range of functions and tools for data representation, analysis, and display (Berry et al., 2021). R also incorporates original programming concepts, including object-oriented programming, which users can use transparently (Micheaux et al., 2013). When it comes to anthropology, Purzycki and Jamieson-Lane (2017) developed the AnthroTools R package for cross-cultural ethnographic data analysis, making these methods more accessible. Baxter and Cool (2010) demonstrated the application of R in archaeology, particularly in correspondence analysis. Through these applications, it has indicated its versatility and potential for further use in anthropology.

In order to further unleash the potential of R programming, this paper will introduce in detail how to use R programming as a tool for somatoploting. The paper will present the code for the somatoploting and its components, insert x and y data into the code, and execute the program with the given code to get the somatochart. Further, the paper elaborates on how to represent a variation between two populations. Overall, the paper will guide the reader to somatoplot using R programming.

Instaling RStudio Program

RStudio is a specialised Integrated Development Environment (IDE) designed exclusively to work with the R programming language. (Jourdan, 2020) Prior to installing RStudio, we must have R installed on our computer. R software can be obtained from the official Comprehensive R Archive Network (CRAN) website at https://cran.r-project.org.

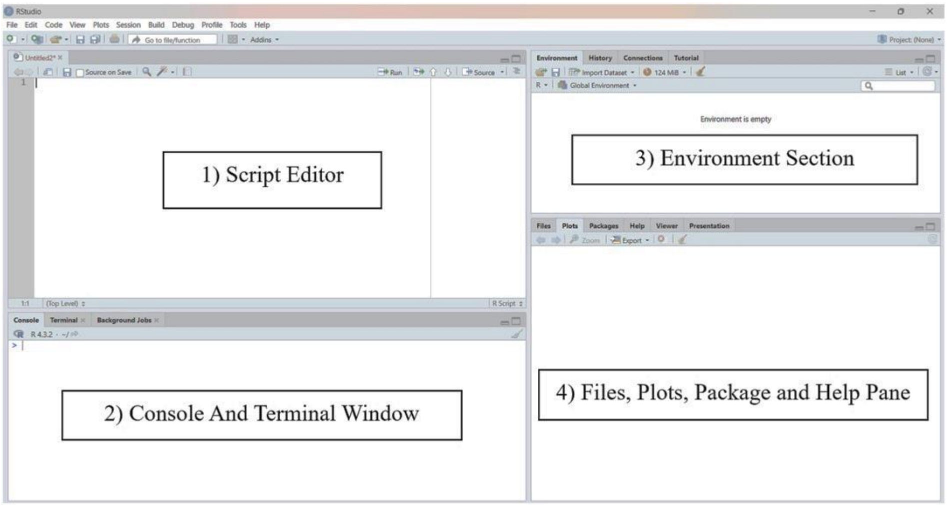

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled2*
Source on Save
Run
Source
Environment History Connections Tutorial
Import Dataset
Global Environment
Environment is empty
Files Plots Packages Help Viewer Presentation
Zoom Export
Console Terminal Background Jobs
R 4.3.2
1) Script Editor
2) Console And Terminal Window
3) Environment Section
4) Files, Plots, Package and Help Pane

Figure 2. RStudio Interface

One must adhere to the installation guidelines for one's specific operating system (Windows/Linux/macOS). After the installation of R, we proceed with the installation of RStudio. A suitable edition of RStudio needs to be obtained from our operating system, which is available at the Posit website at https://posit.co/download/rstudio-desktop/. Posit is an open-source data science which is maintaining the RStudio. Once downloaded, run it per the on-screen instructions to complete the installation. Once installed, the program opens with an RStudio Interface.

Figure 2 shows the RStudio interface. The RStudio interface has four basic windows. First, the script editor allows for writing and creating functional programs with functions and codes. The second one is the console and terminal window, which displays written code output. The environment section is the third window that displays variables, the history section shows past transactions, and the connections section shows server or web links. The last window, the files, plots, package, and help pane, shows plots, packages show library downloads and libraries, and the help section provides help. The viewer window displays maps and complex drawings, which can be adjusted using the settings menu (Kaya et al., 2019). As we have understood the basics of the RStudio interface, let us understand the script code we will use to execute somatoploting.

The Script Code for Somatoploting

In order to execute the program, let us briefly understand the command lines of the script. We need to be aware of some primary rules of R language. Firstly, R is case-sensitive. This means that 'height' and 'Height' are considered different variables. Secondly, lines starting with '#' are ignored by R and are used to explain code or comment. Thirdly, the operator '<-' assigns values to variables. For example, x <- 10 assigns the value 10 to the variable x. Lastly, R can run a single command and multiple commands at once. So, it is essential to run all script codes simultaneously for our somatoploting execution. Our script code consists of 22 lines of commands, including the comments. Let us briefly understand each of the code lines.

	Code lines	Description of Code line
1	x <- c(0,-6,6,0,-4.5,4.5,0)	x and y are variables here. Any variable is defined using <- after the variable symbol. x and y are arrays of numbers. Array means a row of numbers. These allow us to outline the reuleaux triangle of 2-D somatochart. So 'c' is the combination function in (number 1, number 2, number 3, number 4), the format to store a few numbers inside x or y.
2	y <- c(12,-6,-6,0,4.5,4.5,-7.5)	
3	par(mar = c(2,0,0,2), mgp = c(2,1,0))	The 'par()' function allows us to retrieve and alter the collection of graphics parameters for the current graphics device. The par(mar = c(bottom, left, top, right)) syntax specifies the size of the margins using the variables bottom, left, top, and right. The syntax par(mgp=c(axis title, axis labels, and axis line) sets the distance away from 0 of the axis title, axis labels, and axis line.
4	plot(x,y,pch = 20, xlab = " ", ylab = " ", xlim = c(-8,8), ylim = c(-10,16), las = 1, col = "white", axes = F)	The plot (x,y) function generates a scatterplot using two vectors, x, and y, where the x vector represents the horizontal values of the points, and the y vector represents the vertical values. (Phillips, 2017) 'pch' or plot character refers to a graphical argument utilised to indicate the shape of points. The arguments 'xlab' and ylab show labels for the x and y axes (which are not given in our case), respectively. Moreover, xlim and ylim propose arguments for the x-axis and y-axis limits. The 'las' or label for axis style argument helps us define the alignment of the text incorporated into a plot. The available choices include aligned parallel to the axis (the default, las = 0), consistently horizontal (las = 1), consistently perpendicular to the axis (las = 2), and consistently vertical (las = 3). The 'col' argument allows one to choose the colour of lines. Lastly, the 'axes' argument allows for the border of the whole plot. Here, 'F' represents false as the plot does not need borders.
5	axis(4, las = 1, yaxp = c(-10,16,13), cex.axis=0.8)	The axis() function assists us in inserting the axis lines into the plot. 4 or 1 post bracket represents the side of the plot. The 'las' argument helps the orientation of the labels. The xaxp (X-axis) and yaxp (Y-axis) arguments enable the customisation of the tick positions and the number of divisions on each axis. It is achieved by specifying vectors in the format c(start, end, number_regions). Finally, the 'cex.axis'

6	axis(1, xaxp = c(-8, 8, 16), cex.axis=0.8)	determines the scale of the text, number, or symbols, which should be scaled relative to the default. Here, ‘1’ is the default, 1.5 is 50% larger and 0.5 is 50% smaller than the default.
7	# Segmentes	This is optional. It signifies that the following code lines are related to segments.
8	segments(x0 = 0, y0=-7.5, x1 = 0, y1 = 12, lty = 2)	The ‘segments()’ function is used to draw line segments to draw the diagonals for our plot. (x0,y0) is the starting points and (x1,y1) is the endpoints of these diagonals. The argument ‘lty’ or line type can be varied. The representation can be expressed using either textual terms ("blank," "solid," "dashed," "dotted," "dotdash," "longdash," "twodash") or numerical values (0, 1, 2, 3, 4, 5, 6). The lty = "solid" is equivalent to lty=1.
9	segments(x0 = -6, y0=-6, x1 = 4.5, y1 = 4.5, lty = 2)	
10	segments(x0 = 6, y0=-6, x1 = -4.5, y1 = 4.5, lty = 2)	
11	# text	This is optional. It signifies that the following code lines are related to the text in the plot.
12	par(family = "serif")	The ‘par()’ function allows us to retrieve and alter the collection of graphics parameters for the current graphics device. The ‘family’ argument helps change the font style of all texts in the graph. Here, we are using a ‘serif’ font. We can change it to suit our convenience.
13	text(0,13,"MESOMORPH", cex = 0.6, font = 2)	The text() is used to place our text in the plot. The first two numbers in the bracket are x and y coordinates for the location of the text, then the text in double quotes, and ‘cex’ is the number used to increase or decrease(scale) the text. The ‘font’ argument is an integer which specifies which font to use for text. If possible, device drivers arrange that 1 corresponds to plain text (the default), 2 to bold text, 3 to italic and 4 to bold italic.
14	text(-6,-8,"ENDOMORPH", cex = 0.6, font = 2)	
15	text(6,-8,"ECTOMORPH", cex = 0.6, font = 2)	
16	xspline(y = c(-6, 4.5, 12), x = c(-6, -4.5, 0), shape = -1, lty = 2)	‘xspline' is used to draw the curved lines for the triangular shape. x and y are an array of points to list out many points in short. The shape lies between -1 to 1, and lty is the line type (discussed previously).
17	xspline(y = c(-6, -7.5,-6), x = c(-6, 0, 6), shape = -1, lty = 2)	
18	xspline(y = c(-6, 4.5, 12), x = c(6, 4.5, 0), shape = -1, lty = 2)	
19	# Inserting new data points	This is optional. It is used to show that the following code lines are related to coordinates used to insert somatotypes for somatoploting.
20	new_x <- c()	Like the 1st two variables, x and y, new_x and new_y are two variables, also array with the same number of elements for each. These are the x and y coordinates of the somatotypes of samples. The detailed process has been discussed in the following parts.
21	new_y <- c()	
22	points(new_x, new_y, pch = 20, col-“black”))	To make a scatter plot, we generally use the function "plot." However, here, we use points() as we want to add the points to the existing chart. The ‘pch’ helps shape these points, and ‘col’ provides colour.

Preparing the Data

In order to execute the code correctly, we need to prepare our data priorly. Let us take field-collected data for such execution. The first author collected first-hand anthropometric data using the International Society Advancement of Kinanthropometry (ISAK) protocol guidelines (Esparza-Ros et al., 2019) from 34 school-going boys aged between 10 and 12 at Shindikurbet, Karnataka. Informed consent was obtained from the parents and school authorities. Before this, the measurements and their purposes were shown to all students, and only those who volunteered were measured. The parameters of stature height using a stadiometer, weight using a calibrated weighing scale, skinfolds (biceps, triceps, subscapular, supraspinale, and calf) using CESCORF skinfold calliper, girths (upper arm relaxed, upper arm flexed and tensed and calf) using CESCORF anthropometric tape and breadths (humerus and femur) using CESCORF small sliding calliper were measured according to protocol guidelines.

Further, endomorphy, ectomorphy, and mesomorphy values are calculated according to the Heath and Carter method somatotype rating (Chakrabarti & Adhikari, 2023; Heath & Carter, 1967). Further, the x and y coordinate values were calculated (Bawari et al., 2023) and rounded to two decimals. Table 1 shows the data of these students and calculated values.

Table 1. Somatypes of 34 boys

Age	**Type of Student 1) Day Scholar 2) Hosteler**	**Endomorphy**	**Mesomorphy**	**Ectomorphy**	**X**	**Y**
12	1	4.671	3.494	3.472	-1.19	-1.15
11	1	4.706	2.579	4.663	-0.04	-4.21
10	1	4.447	3.449	3.736	-0.71	-1.29
11	1	7.031	3.220	2.911	-4.11	-3.50
11	1	3.346	3.637	3.628	0.28	0.30
10	1	4.085	3.064	4.483	0.39	-2.44
10	1	2.466	2.970	5.174	2.70	-1.70
12	1	2.183	3.275	4.039	1.85	0.33
11	1	7.291	5.074	1.302	-5.98	1.55
11	1	6.312	4.750	1.977	-4.33	1.21
10	1	2.842	2.085	5.641	2.79	-4.31
12	1	5.154	3.052	4.353	-0.80	-3.40
12	1	6.827	4.517	1.868	-4.95	0.34
12	1	6.915	3.340	2.864	-4.05	-3.10
10	1	2.766	2.693	6.127	3.36	-3.51
10	1	5.053	4.813	2.180	-2.87	2.39
10	1	2.194	2.844	4.900	2.70	-1.41
10	1	2.629	3.454	5.078	2.44	-0.80
10	1	3.952	3.296	4.248	0.29	-1.61
10	1	1.990	3.456	5.196	3.20	-0.27
12	1	2.456	2.650	5.856	3.39	-3.01
11	1	5.236	3.625	3.637	-1.59	-1.62
12	1	2.550	3.090	4.664	2.11	-1.03
12	2	5.504	3.955	3.590	-1.91	-1.18
12	2	6.360	3.430	3.326	-3.03	-2.83
12	2	4.714	3.430	4.297	-0.41	-2.15

12	2	3.098	3.828	4.823	1.72	-0.26
10	2	5.656	4.293	2.407	-3.24	0.52
11	2	3.550	3.580	4.487	0.93	-0.88
10	2	5.857	4.126	3.531	-2.32	-1.14
11	2	5.547	3.858	2.895	-2.65	-0.73
10	2	3.819	3.076	4.185	0.36	-1.85
10	2	3.281	2.847	4.843	1.56	-2.43
10	2	4.407	4.575	3.388	-1.01	1.36

As we have x and y coordinates calculated, our subsequent work is to transpose these x and y coordinate values into a single line separated by a comma(,). It can be done manually or with an Excel function, =TEXTJOIN(", ", TRUE, G1:G34) (where G1:G34 is the column we need to convert to a serrated list, ", " indicates how we want to separate the list). Using such an Excel function, the values of the x and y coordinates in Table 1 have been presented below.

Values of x coordinates = -1.19, -0.04, -0.71, -4.11, 0.28, 0.39, 2.7, 1.85, -5.98, -4.33, 2.79, -0.8, -4.95, -4.05, 3.36, -2.87, 2.7, 2.44, 0.29, 3.2, 3.39, -1.59, 2.11, -1.91, -3.03, -0.41, 1.72, -3.24, 0.93, -2.32, -2.65, 0.36, 1.56, -1.01

Values of y coordinates = -1.15, -4.21, -1.29, -3.5, 0.3, -2.44, -1.7, 0.33, 1.55, 1.21, -4.31, -3.4, 0.34, -3.1, -3.51, 2.39, -1.41, -0.8, -1.61, -0.27, -3.01, -1.62, -1.03, -1.18, -2.83, -2.15, -0.26, 0.52, -0.88, -1.14, -0.73, -1.85, -2.43, 1.36

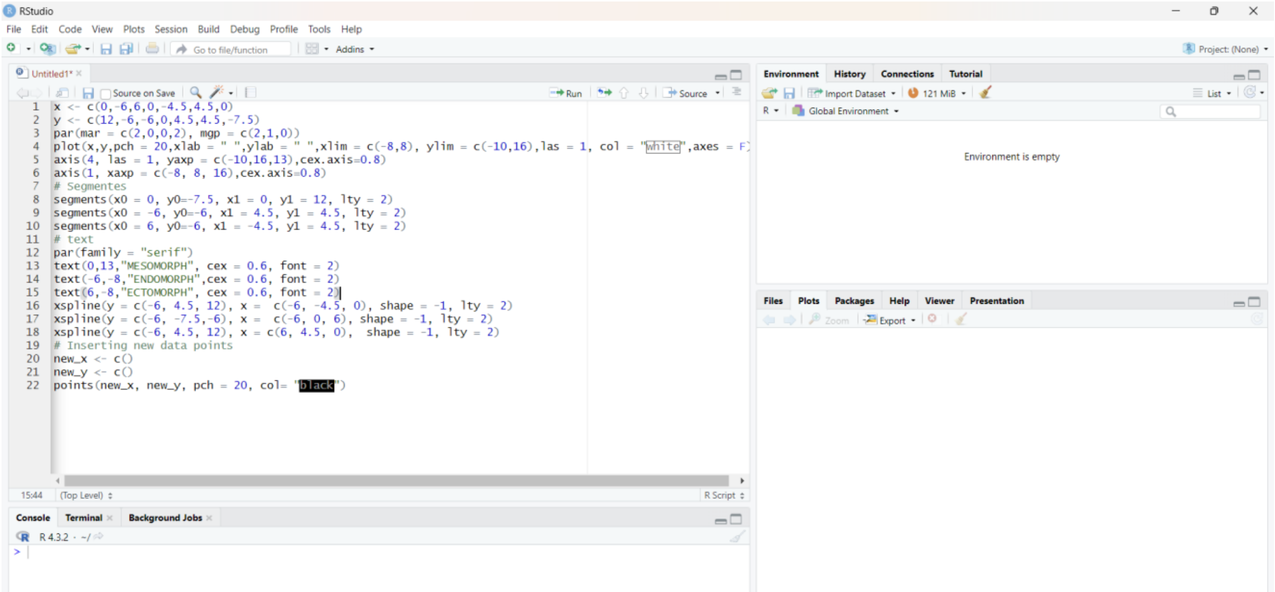
Once these values in the above format are ready, our data is ready to get executed into R script code.

Execution of the Code

As our data is ready, we will insert the above data to perform somatoploting in the RStudio program to get our somatochart as an output file. It can be done in the following steps.

a) Step 1: Run the RStudio program on our computer.

b) Step 2: Type or Paste all the 22-line code in the script editor window of the RStudio (Figure 3).

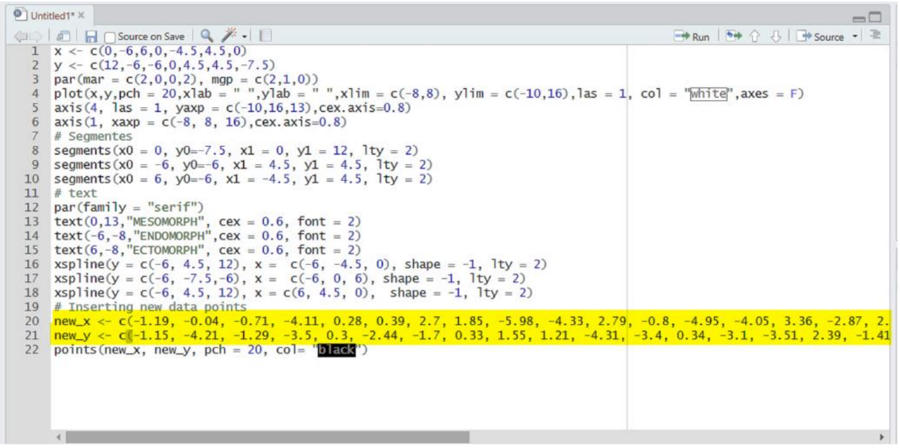


```
1  x <- c(0,-6,6,0,-4.5,4.5,0)
2  y <- c(12,-6,-6,0,4.5,4.5,-7.5)
3  par(mar = c(2,0,0,2), mgp = c(2,1,0))
4  plot(x,y,pch = 20,xlab = " ",ylab = " ",xlim = c(-8,8), ylim = c(-10,16),las = 1, col = "white",axes = F)
5  axis(4, las = 1, yaxp = c(-10,16,13),cex.axis=0.8)
6  axis(1, xaxp = c(-8, 8, 16),cex.axis=0.8)
7  # Segmentes
8  segments(x0 = 0, y0=-7.5, x1 = 0, y1 = 12, lty = 2)
9  segments(x0 = -6, y0=-6, x1 = 4.5, y1 = 4.5, lty = 2)
10 segments(x0 = 6, y0=-6, x1 = -4.5, y1 = 4.5, lty = 2)
11 # text
12 par(family = "serif")
13 text(0,13,"MESOMORPH", cex = 0.6, font = 2)
14 text(-6,-8,"ENDOMORPH",cex = 0.6, font = 2)
15 text(6,-8,"ECTOMORPH", cex = 0.6, font = 2)
16 xspline(y = c(-6, 4.5, 12), x =  c(-6, -4.5, 0), shape = -1, lty = 2)
17 xspline(y = c(-6, -7.5,-6), x =  c(-6, 0, 6), shape = -1, lty = 2)
18 xspline(y = c(-6, 4.5, 12), x = c(6, 4.5, 0),  shape = -1, lty = 2)
19 # Inserting new data points
20 new_x <- c()
21 new_y <- c()
22 points(new_x, new_y, pch = 20, col= "black")
```

```
>
```

Figure 3. Script Code pasted in the Script editor window

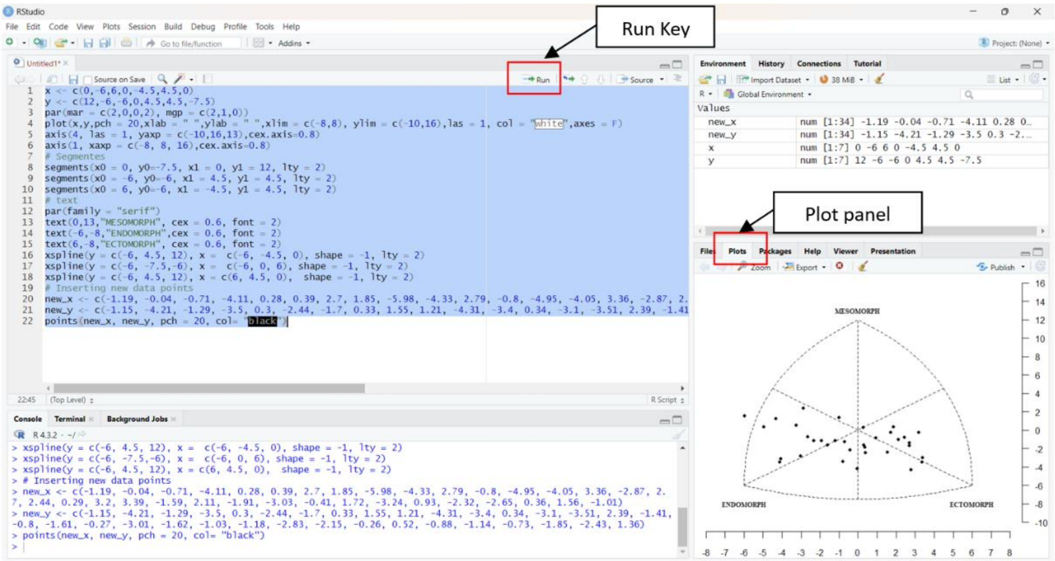
c) Step 3: Paste the x and y coordinate values in line number 20 (new_x) and 21 (new_y), respectively, between the brackets (Figure 4). Care should be taken that the values are between the brackets and that there is no extra comma after the last value.



```
1  x <- c(0,-6,6,0,-4.5,4.5,0)
2  y <- c(12,-6,-6,0,4.5,4.5,-7.5)
3  par(mar = c(2,0,0,2), mgp = c(2,1,0))
4  plot(x,y,pch = 20,xlab = " ",ylab = " ",xlim = c(-8,8), ylim = c(-10,16),las = 1, col = "white",axes = F)
5  axis(4, las = 1, yaxp = c(-10,16,13),cex.axis=0.8)
6  axis(1, xaxp = c(-8, 8, 16),cex.axis=0.8)
7  # Segmentes
8  segments(x0 = 0, y0=-7.5, x1 = 0, y1 = 12, lty = 2)
9  segments(x0 = -6, y0=-6, x1 = 4.5, y1 = 4.5, lty = 2)
10 segments(x0 = 6, y0=-6, x1 = -4.5, y1 = 4.5, lty = 2)
11 # text
12 par(family = "serif")
13 text(0,13,"MESOMORPH", cex = 0.6, font = 2)
14 text(-6,-8,"ENDOMORPH",cex = 0.6, font = 2)
15 text(6,-8,"ECTOMORPH", cex = 0.6, font = 2)
16 xspline(y = c(-6, 4.5, 12), x =  c(-6, -4.5, 0), shape = -1, lty = 2)
17 xspline(y = c(-6, -7.5,-6), x =  c(-6, 0, 6), shape = -1, lty = 2)
18 xspline(y = c(-6, 4.5, 12), x = c(6, 4.5, 0),  shape = -1, lty = 2)
19 # Inserting new data points
20 new_x <- c(-1.19, -0.04, -0.71, -4.11, 0.28, 0.39, 2.7, 1.85, -5.98, -4.33, 2.79, -0.8, -4.95, -4.05, 3.36, -2.87, 2.
21 new_y <- c(-1.15, -4.21, -1.29, -3.5, 0.3, -2.44, -1.7, 0.33, 1.55, 1.21, -4.31, -3.4, 0.34, -3.1, -3.51, 2.39, -1.41
22 points(new_x, new_y, pch = 20, col= "black")
```

Figure 4. X and Y coordinates added in lines 20 and 21, respectively

d) Step 4: Select all the script code in the script editor window and press the 'Run' key on the top right of the same window. Note that it is vital to select the whole text at once and execute it; otherwise, the somatochart will not appear. Once the code runs and is executed, our somatochart will appear in the fourth window, i.e., the files, plots, package, and help pane under the plots panel (Figure 5).



```
1  x <- c(0,-6,6,0,-4.5,4.5,0)
2  y <- c(12,-6,-6,0,4.5,4.5,-7.5)
3  par(mar = c(2,0,0,2), mgp = c(2,1,0))
4  plot(x,y,pch = 20,xlab = " ",ylab = " ",xlim = c(-8,8), ylim = c(-10,16),las = 1, col = "white",axes = F)
5  axis(4, las = 1, yaxp = c(-10,16,13),cex.axis=0.8)
6  axis(1, xaxp = c(-8, 8, 16),cex.axis=0.8)
7  # Segmentes
8  segments(x0 = 0, y0=-7.5, x1 = 0, y1 = 12, lty = 2)
9  segments(x0 = -6, y0=-6, x1 = 4.5, y1 = 4.5, lty = 2)
10 segments(x0 = 6, y0=-6, x1 = -4.5, y1 = 4.5, lty = 2)
11 # text
12 par(family = "serif")
13 text(0,13,"MESOMORPH", cex = 0.6, font = 2)
14 text(-6,-8,"ENDOMORPH",cex = 0.6, font = 2)
15 text(6,-8,"ECTOMORPH", cex = 0.6, font = 2)
16 xspline(y = c(-6, 4.5, 12), x =  c(-6, -4.5, 0), shape = -1, lty = 2)
17 xspline(y = c(-6, -7.5,-6), x =  c(-6, 0, 6), shape = -1, lty = 2)
18 xspline(y = c(-6, 4.5, 12), x = c(6, 4.5, 0),  shape = -1, lty = 2)
19 # Inserting new data points
20 new_x <- c(-1.19, -0.04, -0.71, -4.11, 0.28, 0.39, 2.7, 1.85, -5.98, -4.33, 2.79, -0.8, -4.95, -4.05, 3.36, -2.87, 2.
21 new_y <- c(-1.15, -4.21, -1.29, -3.5, 0.3, -2.44, -1.7, 0.33, 1.55, 1.21, -4.31, -3.4, 0.34, -3.1, -3.51, 2.39, -1.41
22 points(new_x, new_y, pch = 20, col= "black")
```

```
> xspline(y = c(-6, 4.5, 12), x =  c(-6, -4.5, 0), shape = -1, lty = 2)
> xspline(y = c(-6, -7.5,-6), x =  c(-6, 0, 6), shape = -1, lty = 2)
> xspline(y = c(-6, 4.5, 12), x = c(6, 4.5, 0),  shape = -1, lty = 2)
> # Inserting new data points
> new_x <- c(-1.19, -0.04, -0.71, -4.11, 0.28, 0.39, 2.7, 1.85, -5.98, -4.33, 2.79, -0.8, -4.95, -4.05, 3.36, -2.87, 2.
7, 2.44, 0.29, 3.2, 3.39, -1.59, 2.11, -1.91, -3.03, -0.41, 1.72, -3.24, 0.93, -2.32, -2.65, 0.36, 1.56, -1.01)
> new_y <- c(-1.15, -4.21, -1.29, -3.5, 0.3, -2.44, -1.7, 0.33, 1.55, 1.21, -4.31, -3.4, 0.34, -3.1, -3.51, 2.39, -1.41,
-0.8, -1.61, -0.27, -3.01, -1.62, -1.03, -1.18, -2.83, -2.15, -0.26, 0.52, -0.88, -1.14, -0.73, -1.85, -2.43, 1.36)
> points(new_x, new_y, pch = 20, col= "black")
> 
```

```
Values
new_x     num [1:34] -1.19 -0.04 -0.71 -4.11 0.28 0…
new_y     num [1:34] -1.15 -4.21 -1.29 -3.5 0.3 -2.…
x         num [1:7] 0 -6 6 0 -4.5 4.5 0
y         num [1:7] 12 -6 -6 0 4.5 4.5 -7.5
```

Figure 5. Running the Code and 2-D Somatochart appearance

e) Step 5: The final stage is exporting the somatochart for further use. It can be copied and used for other purposes or saved as an image in PNG, jpeg, SVG, tiff, BMP, or metafile format. It can also be exported as a PDF and saved on the computer. The export option is available in the fourth window. The image's aspect ratio can be adjusted for clarity using the options given (Figure 6)

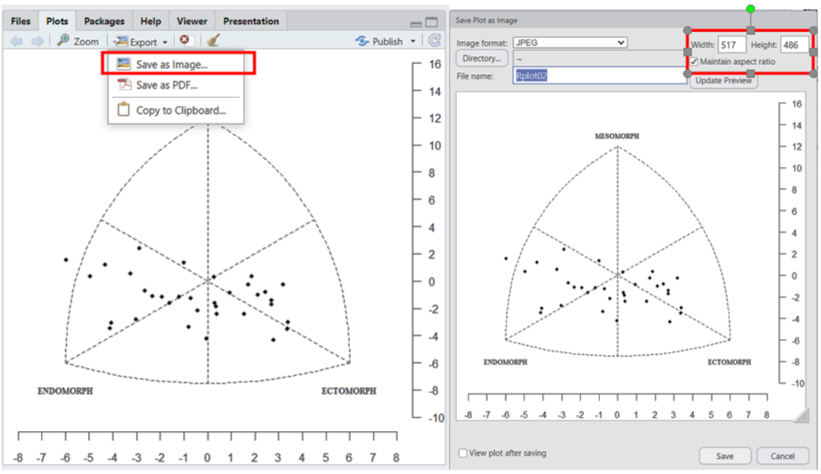


Image format: JPEG

Directory... ~

File name: Rplot02

Width: 517 Height: 486

☑ Maintain aspect ratio

□ View plot after saving

Figure 6. Exporting the 2-D Somatachart

Following the above five steps, one can somatoplot and produce quick and refined somatocharts. These somatocharts can be further furnished with labels accordingly through various software.

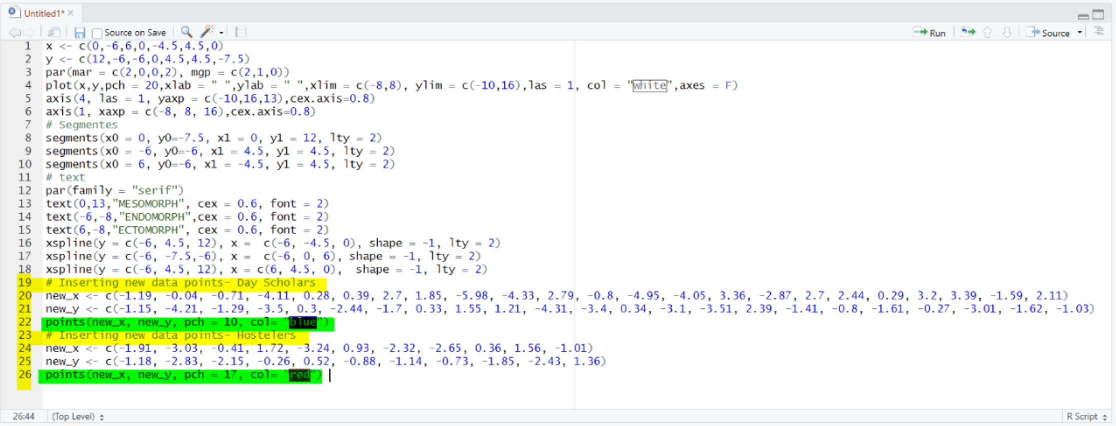
Somatoploting of two samples

In most cases, the process of somatoploting and somatocharts are used to compare populations or samples (Chakrabarti & Adhikari, 2023; Kabir et al., 2023; Manna et al., 2022; Tadesse et al., 2022). Understanding its potential, we will discuss using R programming to somatoplot two samples. We will consider the same sample of 34 boys and classify them as Day scholars (n=23) and Hostelers (n=11). We must adhere to the same process of preparing the data and transposing the x and y coordinates. Table 2 shows these groups' transposed x and y coordinate values.

Table 2. Transposed x and y coordinate values of Day scholars and Hostelers

X and Y Coordinates of Day Scholars (n= 23)	
X	-1.19, -0.04, -0.71, -4.11, 0.28, 0.39, 2.7, 1.85, -5.98, -4.33, 2.79, -0.8, -4.95, -4.05, 3.36, -2.87, 2.7, 2.44, 0.29, 3.2, 3.39, -1.59, 2.11
Y	-1.15, -4.21, -1.29, -3.5, 0.3, -2.44, -1.7, 0.33, 1.55, 1.21, -4.31, -3.4, 0.34, -3.1, -3.51, 2.39, -1.41, -0.8, -1.61, -0.27, -3.01, -1.62, -1.03
X and Y Coordinates of Hostelers (n=11)	
X	-1.91, -3.03, -0.41, 1.72, -3.24, 0.93, -2.32, -2.65, 0.36, 1.56, -1.01
Y	-1.18, -2.83, -2.15, -0.26, 0.52, -0.88, -1.14, -0.73, -1.85, -2.43, 1.36

After we have these values, we will execute them using the same steps we did in the last section discussed previously. Nevertheless, we will modify the 22-line code. We add command lines from 19 to 22 again into the script and re-label it according to our groups (Line 19 and Line 23) (Figure 7). Similar to the other steps, we will add our x and y coordinate values, respectively.



```
1  x <- c(0,-6,6,0,-4.5,4.5,0)
2  y <- c(12,-6,-6,0,4.5,4.5,-7.5)
3  par(mar = c(2,0,0,2), mgp = c(2,1,0))
4  plot(x,y,pch = 20,xlab = " ",ylab = " ",xlim = c(-8,8), ylim = c(-10,16),las = 1, col = "white",axes = F)
5  axis(4, las = 1, yaxp = c(-10,16,13),cex.axis=0.8)
6  axis(1, xaxp = c(-8, 8, 16),cex.axis=0.8)
7  # Segmentes
8  segments(x0 = 0, y0=-7.5, x1 = 0, y1 = 12, lty = 2)
9  segments(x0 = -6, y0=-6, x1 = 4.5, y1 = 4.5, lty = 2)
10 segments(x0 = 6, y0=-6, x1 = -4.5, y1 = 4.5, lty = 2)
11 # text
12 par(family = "serif")
13 text(0,13,"MESOMORPH", cex = 0.6, font = 2)
14 text(-6,-8,"ENDOMORPH",cex = 0.6, font = 2)
15 text(6,-8,"ECTOMORPH", cex = 0.6, font = 2)
16 xspline(y = c(-6, 4.5, 12), x =  c(-6, -4.5, 0), shape = -1, lty = 2)
17 xspline(y = c(-6, -7.5,-6), x =  c(-6, 0, 6), shape = -1, lty = 2)
18 xspline(y = c(-6, 4.5, 12), x = c(6, 4.5, 0),  shape = -1, lty = 2)
19 # Inserting new data points- Day Scholars
20 new_x <- c(-1.19, -0.04, -0.71, -4.11, 0.28, 0.39, 2.7, 1.85, -5.98, -4.33, 2.79, -0.8, -4.95, -4.05, 3.36, -2.87, 2.7, 2.44, 0.29, 3.2, 3.39, -1.59, 2.11)
21 new_y <- c(-1.15, -4.21, -1.29, -3.5, 0.3, -2.44, -1.7, 0.33, 1.55, 1.21, -4.31, -3.4, 0.34, -3.1, -3.51, 2.39, -1.41, -0.8, -1.61, -0.27, -3.01, -1.62, -1.03)
22 points(new_x, new_y, pch = 10, col= "blue")
23 # Inserting new data points- Hostelers
24 new_x <- c(-1.91, -3.03, -0.41, 1.72, -3.24, 0.93, -2.32, -2.65, 0.36, 1.56, -1.01)
25 new_y <- c(-1.18, -2.83, -2.15, -0.26, 0.52, -0.88, -1.14, -0.73, -1.85, -2.43, 1.36)
26 points(new_x, new_y, pch = 17, col= "red")
```

Figure 7. Inserting x and y values of Day scholars and Hostelers

One of the critical modifications we can make to differentiate the points of the two groups on the somatochart is to change the shape and colours of points. This modification can be quickly done on the command code of line 22 and line 26 (Figure 7). The points function's 'pch' or plot character refers to a graphical argument utilised to indicate the shape of points. The 'pch' value can be between 0 and 25 (Ganguly, 2023). The different shapes associated with the number are presented in Figure 8.

Similarly, the 'col' graphical argument allows colour to be allotted to these points accordingly. Observe lines 22 and 26 in Figure 7, where the points of Day scholars were allotted blue, and hostelers were given red. We can assign these modifications according to our convenience and choice. Figure 9 is the output somatochart of the somatotypes of day scholars and hostelers when we run the whole script code, as mentioned in Figure 7.

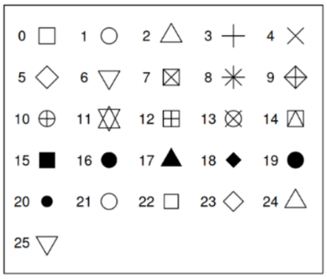


Figure 8. 'pch' values associated shapes. Accessed at https://r-charts.com/base-r/pch-symbols/

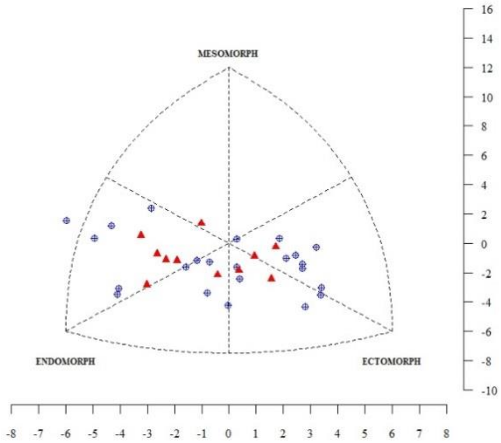


Figure 9. Somatochart representing Day scholars (blue) and Hostelers (red) somatotypes

Discussion and Conclusion

Data visualisation, facilitated by computer graphics, aids in understanding complex information, aids in fact-explaining and guiding action in various fields and benefits innovative presentation methods (Sadiku et al., 2016). The process of somatoploting and its product – the somatocharts as data analysing and data visualisation tools are vital in various studies of human physiology, anthropometry, human growth and development, nutrition and health. This work aims to demonstrate using R programming, an open-source software, for somatoploting and generating somatocharts. Open-source software such as R programming offers accessible data processing, analysis, and presentation, making it easy to implement and achieve budget savings for students, researchers, numerous institutions and universities (Kaya et al., 2019). Further research can be done to furnish the code into Excel sheets or mobile applications and make it easier and simpler to use.

References

Anand, S. (2022). R programming language - Introduction and basics. Blogs & Updates on Data Science, Business Analytics, AI Machine Learning. Retrieved January 22, 2024, from https://www.analytixlabs.co.in/blog/r-programming-language-basics/

Bawari, B., Adhikari, R., Easow, J., Pullinger, S.A. (2023). Somatotype and Body Composition of Indian Male and Female Swimmers, and their Relationship to Countermovement Jump Performance. *International Journal of Kinanthropometry*, 3(2): 84–95. https://doi.org/10.34256/ijk23210

Baxter, M.J., & Cool, H.E.M. (2010). Correspondence analysis in R for archaeologists: an educational account. *Archeologia e Calcolatori*, (21): 211-228.

Berry, K.J., Kvamme, K.L., Johnston, J.E., Mielke, P.W., Berry, K.J., Kvamme, K. L., Johnston, J.E., & Mielke, P.W. (2021). The R Programming Language. Permutation Statistical Methods with R. Springer, Cham. https://doi.org/10.1007/978-3-030-74361-1_2

Chakrabarti, D., & Adhikari, A. (2023). Somatotype Characteristics of Indian National Canoeists and Kayakers. *International Journal of Kinanthropometry*, 3: 118–123. https://doi.org/10.34256/ijk23113

Chiu, C., Ciems, R., Thelwell, M., Bullas, A., & Choppin, S. (2022). Estimating somatotype from a single-camera 3D body scanning system. *European Journal of Sport Science,* 22(8), 1204–1210. https://doi.org/10.1080/17461391.2021.1921041

Esparza-Ros, F., Vaquero-Cristobal, R., & Marfell-Jones, M. (2019). International Standards for Anthropometric Assessment (2019) Restricted Profile. *The International Society for the Advancement of Kinanthropometry*, Spain.

Ganguly, A. (2023). Plot Characters (PCH) in R. *FavTutor*. Plot Characters (PCH) in R (favtutor.com)

Heath, B.H., & Carter, J.L. (1967). A modified somatotype method. *American journal of physical anthropology*, 27(1): 57-74. https://doi.org/10.1002/ajpa.1330270108

Horgan, J.M. (2012). Programming in R. *WIREs. Computational Statistics*, 4(1): 75–84. https://doi.org/10.1002/wics.183

Jourdan, A. (2020). Using R for Multivariate Analysis. *TORUS 1–Toward an Open Resource Using Services: Cloud Computing for Environmental Data,* 129-151. https://doi.org/10.1002/9781119720492.ch10

Kabir, M.S., Yadav, S., Prasad, T., & Biswas, S. (2023). Demographic Insights Into College-Going Students in India: A Morphological Analysis. *International Journal of Kinanthropometry,* 3(2): 69-77. https://doi.org/10.34256/ijk2328

Kaya, E., Agca, M., Adiguzel, F., & Cetin, M. (2019). Spatial data analysis with R programming for environment. *Human and Ecological Risk Assessment: An International Journal*, 25(6): 1521–1530. https://doi.org/10.1080/10807039.2018.1470896

Lafaye de Micheaux, P., Drouilhet, R., Liquet, B., de Micheaux, P.L., Drouilhet, R., & Liquet, B. (2013). Programming in R. The R Software. Fundamentals of Programming and Statistical Analysis, 193-282. https://doi.org/10.1007/978-1-4614-9020-3

Manna, T., Adhikari, S., & Syamal, A.K. (2022). Comparison of physical and anthropometrical parameters of teen-age male rowers, kayakers, canoers and sedentary school children. International Journal of Kinanthropometry, 2(1): 25-30.16. https://doi.org/10.34256/ijk2214

Purzycki, B.G., & Jamieson-Lane, A. (2017). AnthroTools: An R package for cross-cultural ethnographic data analysis. *Cross-Cultural Research*, 51(1): 51-74. https://doi.org/10.1177/1069397116680352

Sadiku, M., Shadare, A.E., Musa, S.M., Akujuobi, C.M., & Perry, R. (2016). Data visualisation. *International Journal of Engineering Research and Advanced Technology (IJERAT),* 2(12): 11-16.

Sheldon, W.A. (1954). Atlas of men, a guide for somatotyping the adult male at all ages. [1st ed.] New York, Harper

Siders, W., & Rue, M. (1992). Reuleaux triangle somatocharts. *Computers in biology and medicine*, 22(5): 363-368. https://doi.org/10.1016/0010-4825(92)90024-H

Singh, S.P., & Mehta, P. (2009). Human Body Measurements: Concepts And Applications. *PHI Learning Pvt. Ltd.*

Tadesse, L., van Heerden, J., & Tefera, O. (2022). Classification of junior Ethiopian football players using anthropometric and physical fitness attributes: Developing a predictive model. *International Journal of Kinanthropometry*, 2(1): 13-24. https://doi.org/10.34256/ijk2213

Toth, T., Michalíková, M., Bednarčíková, L., Zivcak, J., & Kneppo, P. (2014). Somatotypes in sport. *Acta mechanica et automatica*, 8(1): 27-32.

Tuimala, J., & Kallio, A. (2013). R, programming language. *Encyclopedia of Systems Biology,* 1809-1811.

Wang, J., Thornton, J.C., Kolesnik, S., & Pierson Jr, R. N. (2000). Anthropometry in body composition: an overview. *Annals of the New York Academy of Sciences*, 904(1): 317-326. https://doi.org/10.1111/j.1749-6632.2000.tb06474.x

Acknowledgement

The first author receives a fellowship from UGC-India under the NET-JRF Scheme. The authors convey gratitude to UGC for its funding. The authors thank the administration and parents of The Saraswati International School, Shindikurbet, Karnataka, for their support for data collection.

Conflicts of Interest

The authors have no conflicts of interest to declare that they are relevant to the content of this article.

DOI: 10.34256/ijk2417

About the License