

Lenguajes de programación para la estadística: R, Python y Julia.

Programming languages for statistics: R, Python and Julia.

Raymundo Bueno-Rivera¹, Raúl Quevedo García², Jesús Andrés Aguilar Quiroz³, Juan José López Tirado⁴,
Alan Ramírez-Noriega⁵

¹Facultad de Informática Culiacán, Universidad Autónoma de Sinaloa, México.

²Facultad de Informática Culiacán, Universidad Autónoma de Sinaloa, México.

³Facultad de Informática Culiacán, Universidad Autónoma de Sinaloa, México.

⁴Facultad de Informática Culiacán, Universidad Autónoma de Sinaloa, México.

⁵Facultad de Ingeniería Mochis, Universidad Autónoma de Sinaloa, México.

Autor de Correspondencia: Raymundo Bueno Rivera, e.raymundo.bueno@uas.edu.mx, ORCID: <https://orcid.org/0009-0001-7742-267X>

Recibido: Enero 2025, **Aceptado:** Marzo 2025, **Publicado:** Mayo 2025

Resumen:

La estadística y la probabilidad forman parte del área de matemáticas, y permiten llevar a cabo el análisis de datos, implementando metodologías desde la selección, recolección, tratamiento y presentación de resultados. En este artículo se presenta una comparación del uso de los lenguajes de programación R, Python y Julia como herramientas de apoyo en estas áreas. Esta comparación se llevó a cabo utilizando conjuntos de datos (datasets) para hacer cálculos estadísticos y probabilísticos con cada uno de los lenguajes, lo que permitió evaluar los procedimientos desde la apreciación de los autores, esta evaluación se realizó utilizando una rúbrica enfocada en aspectos cualitativos, los resultados obtenidos se compararon a partir de una escala de Likert con lo que se concluyó que el mejor lenguaje como herramienta de apoyo para esta área es R.

Palabras Clave: Probabilidad y estadística, lenguaje R, Python, lenguaje Julia, comparación de lenguajes de programación.

Abstract:

Probability and statistics is an area from Mathematics, its main purpose is to take over data analysis by applying very specific methods to activities like data selection, data gathering, data treatment and results showing. In this article is shown a comparison of three different programming languages as a support tool for this area, these languages are Python, R and Julia. This study was taken over with different data sets to develop some statistics and probability operations with each of the programming languages, this allowed to evaluate the procedures from the perspective of the authors. This evaluation was done using rubric oriented in qualitative aspects. The obtained results were compared using a Likert scale in which the conclusion was that R is the best language to use as a support tool for this area.

Index Terms: Statistics and probability, R language, Python, Julia language, comparison of programming languages.

1. Introducción

En el estudio y análisis de fenómenos físicos, sociales, químicos y de cualquier índole es de bastante relevancia la aplicación de la estadística, ya que esta es una rama que estudia las formas de recolectar los datos, ordenarlos, mostrarlos e interpretarlos, esto con la finalidad de sacar conclusiones de ellos [1].

La misma estadística se puede dividir en dos grandes categorías, descriptiva e inferencial [2], donde la primera es un poco más intuitiva y se enfoca primordialmente con técnicas para recolectar, ordenar y mostrar la información a través de gráficas o tablas, de tal manera que su representación sea significativa y permita observar aspectos relevantes de los datos estadísticos, o bien muestre un resumen de estos.

Mientras que la segunda permite determinar generalidades o un comportamiento de una población, a partir de datos obtenidos de una muestra representativa de la misma. Donde se entiende como muestra a una parte de la población [3]. Para lograr esto se deben utilizar correctas técnicas de muestreo, hacer cálculos de medidas de tendencia central y de dispersión, y que estas sean de apoyo para definir y estimar el comportamiento de la población.

Pero entonces, ¿Al estimar el comportamiento de una población significa que esto va a suceder con certeza?, en este caso solo se puede determinar el comportamiento con un grado o porcentaje de certeza, la ciencia encargada de determinar eso se le llama probabilidad, si bien está fuertemente ligada con la estadística, esta última es la encargada de la recolección de los datos mientras que la probabilidad utiliza esos datos para determinar el grado de certeza de que algo ocurra [4].

Dicho lo anterior y llevando la probabilidad y estadística a un nivel un poco más elevado, es posible utilizarlas para realizar predicciones, que es una tarea común en ciencias de la información. Un ejemplo es el uso de algoritmos de aprendizaje automático, donde el grado de certeza se relaciona con la incertidumbre del algoritmo de que la predicción sea correcta [5].

Lo anterior lleva indiscutiblemente a trabajar con conjuntos de datos cada vez más grandes, lo que aumenta considerablemente el tiempo que demanda realizar esas tareas, entonces, ¿Cuáles son las alternativas para manejar grandes cantidades de información?

Para solucionar ese problema se han desarrollado computadoras y lenguajes de programación que permiten realizar cálculos matemáticos complejos y análisis de grandes cantidades de datos. Entre estos se puede destacar el lenguaje R que fue creado específicamente para solucionar problemas estadísticos, pero este no es el único lenguaje que se utiliza para este ámbito, un lenguaje de propósito general que cuenta con una gran cantidad de módulo de apoyo para la estadística es Python.

El resto de los lenguajes existentes también se pueden utilizar también como apoyo para la probabilidad y estadística, aunque es probable que la complejidad aumente debido a que se necesitan mayor experiencia como programador. Un ejemplo de lenguaje de programación reciente es Julia.

El objetivo de este trabajo es realizar un análisis comparativo entre 3 lenguajes de programación para determinar ventajas y desventajas de su uso al graficar o emplear operaciones básicas de probabilidad y estadística. En el siguiente capítulo se muestra una descripción de cada uno de los lenguajes a comparar, R, Python y Julia.

En el capítulo 3 se describe la metodología que se siguió, se mencionan los conjuntos de datos utilizados. En el capítulo 4 se describe el instrumento de evaluación y los aspectos a evaluar, desde la perspectiva de los autores.

Finalmente, en el capítulo 5 y 6 se discuten los resultados y se mencionan las conclusiones de este comparativo.

2. Trabajos Relacionados

b. *Lenguajes de programación*

2.1.1. Lenguaje R

R es un lenguaje y entorno para la computación estadística y gráficos. Es un proyecto GNU similar al lenguaje y entorno S, desarrollado en los Laboratorios Bell (antes AT&T, ahora Lucent Technologies) por John Chambers y sus colegas. R puede considerarse una implementación diferente de S. Existen algunas diferencias importantes, pero gran parte del código escrito para S se ejecuta sin alteraciones en R [6].

Uno de los puntos fuertes de R es la facilidad con la que se pueden producir gráficos bien diseñados con calidad de publicación, incluyendo símbolos matemáticos y fórmulas cuando sea necesario. Se ha puesto gran cuidado en los valores por defecto para las opciones de diseño menores en gráficos, pero el usuario conserva el control total [6].

R está disponible como Software Libre bajo los términos de la Licencia Pública General GNU de la Fundación para el Software Libre en forma de código fuente. Se compila y ejecuta en una amplia variedad de plataformas UNIX y sistemas similares (incluyendo FreeBSD y Linux), Windows y MacOS [6].

R, al igual que S, está diseñado en torno a un verdadero lenguaje informático, y permite a los usuarios añadir funcionalidades adicionales mediante la definición de nuevas funciones. Gran parte del sistema está escrito en el dialecto R de S, lo que facilita a los usuarios el seguimiento de las decisiones algorítmicas tomadas. Para tareas de cálculo intensivo, se puede enlazar código C, C++ y Fortran e invocarlo en tiempo de ejecución. Los

usuarios avanzados pueden escribir código C para manipular objetos R directamente [6].

2.1.2. Lenguaje Python

Python es un lenguaje de programación que utiliza estructuras de datos de alto nivel, de igual forma permite programar con una estructura orientada a objetos. El intérprete de Python permite hacer referenci⁴a, a instrucciones y funciones utilizadas en C o C++, además de otros lenguajes que puedan ser llamados desde C [7].

Este lenguaje utiliza un intérprete que funciona en distintos sistemas operativos, como lo son Windows, Mac-OS o Linux, esto le permite al usuario seleccionar el aquel sistema operativo que mejor se adapte a sus necesidades.

Otra característica relevante de este lenguaje es el reducido número de líneas de código que se necesitan para desarrollar un programa si se compara con lenguajes como C o C++. Esto se debe a que el intérprete permite utilizar variables sin declararlas en líneas anteriores, lo cual se vuelve muy amigable para programadores novatos.

Finalmente, otro aspecto a destacar de este lenguaje es que existe una gran variedad de módulos que son libres y cuentan con bastante documentación en línea, los cuales están diseñados para realizar tareas específicas como la lectura de documentos, cálculos estadísticos, creación de gráficas, por destacar algunas.

2.1.3. Lenguaje Julia

Julia fue lanzado originalmente en 2012 por Alan Edelman, Stefan Karpinski, Jeff Bezanson y Viral Shah. Es un lenguaje de programación gratuito y de código abierto. Del cual se pueden destacar las siguientes características [8]:

- Julia está desarrollado como un lenguaje de programación de alto rendimiento.
- Julia usa envío múltiple (“multiple dispatch” en inglés), que le permite al programador elegir entre diferentes patrones de programación de acuerdo con la aplicación.
- Julia es un lenguaje de tipo dinámico que se puede usar fácilmente de forma interactiva.
- Julia tiene una sintaxis de alto nivel que es fácil de aprender.
- Julia es un lenguaje de programación con tipos opcionales, cuyos tipos de datos (definidos por el usuario) hacen que el código sea claro y robusto.
- Julia tiene una biblioteca estándar extendida, además, están disponibles numerosos paquetes de terceros.

Estas características lo convierten en un lenguaje de programación único, ya que resuelve el problema de los dos idiomas, fácil entendimiento y no se necesita de ningún otro lenguaje de programación para escribir

código de alto rendimiento. Aunque esto no significa que ocurra automáticamente, es responsabilidad del programador optimizar el código para evitar que genere cuellos de botella, aunque esto puede hacerse directamente en Julia [8].

3. Metodología.

Para llevar a cabo la comparación de estos tres lenguajes de programación en la resolución de problemas de estadística y probabilidad se procedió a seleccionar conjuntos de datos y procesarlos en R, Python y Julia. En la figura 1 y 2 se muestra un esquema de la metodología implementada.

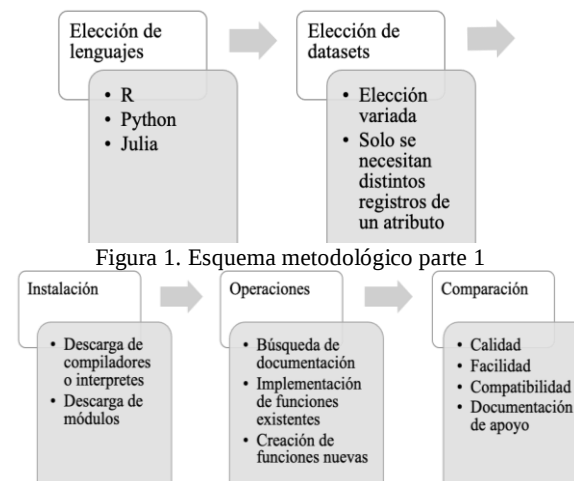


Figura 2. Esquema metodológico parte 2

En el bloque llamado operaciones se realizan las mismas actividades para cada lenguaje de programación, el resultado y el proceso de estas actividades se utilizan para realizar una evaluación cualitativa de calidad, facilidad, compatibilidad y documentación de apoyo.

3.1.1 Cálculos estadísticos

Para desarrollar esta etapa se utilizaron los conjuntos de datos de Iris [9], Superconductivity [10] y Heart Disease [11]. Los cuales son de acceso libre y cuentan con atributos de tipo numérico que permiten llevar a cabo las operaciones estadísticas que se desean probar, las cuales fueron las siguientes:

- Media, media recortada, mediana, varianza y desviación estándar.
- Gráficos de dispersión
- Histograma
- Gráfico de caja y bigotes (boxplot)
- Diagrama de tallos y hojas

En los siguientes apartados se mencionan los procedimientos que se tienen que seguir para cada uno de los lenguajes de programación y los resultados que se obtienen al ejecutar los códigos.

3.1.2 Uso de R

En el caso de este lenguaje de programación no es necesario instalar módulos, solamente se cargan los datos en un objeto y se ejecutan las instrucciones para obtener el resultado que se busque, en la siguiente lista se muestran algunas de las más sencillas:

- MEAN para media.
- MEDIAN para mediana.
- MEAN (datos, trim=valor) donde valor representa un número real entre (0-1) con el porcentaje a recortar.
- MAX(datos)-MIN(datos) calcula el rango de los valores.

3.1.3 Uso de Python

Para los cálculos relevantes de Probabilidad y Estadística como lo son media, mediana, desviación estándar o rango, no es necesario realizar instalaciones de otros módulos, pero sí es necesario declarar la importación de esos módulos al código, para crear objetos y clases que cuenten con esas funciones.

Mientras que para la creación y manejo de gráficos en Python es necesario realizar una instalación extra de módulos específicos. De acuerdo con los tipos de gráficos que se realizaron en esta actividad se instalan los siguientes módulos por medio de la consola de la computadora:

- matplotlib: Para los gráficos comunes como dispersión, histograma, boxplot, barras, etc.
- stemgraphic: Para el diagrama de tallos y hojas

Ambos módulos se tienen que importar en el código del programa.

3.1.4 Uso de Julia

En el caso de uso de Julia es algo muy similar a lo que sucede en R, no hay necesidad de instalar módulos extras para los cálculos básicos o relevantes de probabilidad y estadística, simplemente la ejecución de las instrucciones.

4. Resultados

De manera general los tres lenguajes de programación utilizados mostraron buenas cualidades para trabajar cálculos estadísticos y creación de gráficos, en las siguientes tablas se muestra un resumen de los aspectos evaluados, esto desde la percepción de los autores que realizaron la actividad usando ese lenguaje.

La evaluación de los lenguajes de programación se llevó a cabo con una rúbrica, en la que se evaluaron los atributos de la siguiente lista con una escala de: Excelente, bueno, regular, mal y muy mal.

- Facilidad de instalación
- Documentación de apoyo
- Instalación de plugins o librerías
- Complejidad del lenguaje

E. Configuración de los parámetros de las funciones

F. Calidad de los gráficos

En las Tablas 1, 2 y 3 se muestra la evaluación otorgada en cada atributo y una descripción justificando esa valoración.

Tabla 1. Evaluación de la experiencia usando R.

Atributo	Lenguaje: R
A	Excelente
	El proceso de instalación en Windows y MacOS está bien definido.
B	Excelente
	Bastante documentación proporcionada por R, bastantes blogs de usuarios con asistencia.
C	Excelente
	Es posible realizar la instalación directamente desde la terminal de RStudio.
D	Excelente
	Sintaxis muy sencilla, intuitiva y fácil de aprender. R puede ser un buen primer lenguaje de programación.
E	Excelente
	Existen varios argumentos para cada función, que permiten una excelente personalización de cada gráfico.
F	Excelente
	Los gráficos en R están muy bien definidos, los argumentos de cada función permiten una gran variedad de personalizaciones.

Tabla 2. Evaluación de la experiencia usando Python.

Atributo	Lenguaje: Python
A	Excelente
	Tanto para Windows como en MacOS es un proceso muy sencillo y mecánico.
B	Bueno

Atributo	Lenguaje: Python
	La cantidad de documentación que existe bastante, solo en el gráfico de tallos y hojas fue muy escasa.

Tabla 2. Evaluación de la experiencia usando Python (continuación).

Atributo	Lenguaje: Python
C	Regular
	La instalación es sencilla, pero se necesita conocer el trabajo en consola.
D	Excelente
	No se necesita tener mucho conocimiento del lenguaje para usarlo, ya que simplifica muchos aspectos de diseño.
E	Bueno
	La configuración es sencilla, sin embargo y sobre todo en gráfico no es muy claro y se tiene que realizar varias pruebas para encontrar la configuración deseada.
F	Bueno
	Gráficos muy bien definidos, la especificación depende del programador.

Tabla 3. Evaluación de la experiencia usando Julia.

Atributo	Lenguaje: Julia
A	Excelente
	La instalación de Julia en Windows, Linux y MacOS es bastante sencillo
B	Regular
	La cantidad de documentación en sitios oficiales de las librerías o extensiones esta completa pero falta de actualización para versiones de julia recientes
C	Bueno
	La instalación mediante REPL de Julia hace el proceso de instalación fácil.
D	Excelente

	Posee una sintaxis similar a Python por lo que al usuario principiante puede ser otra opción como lenguaje inicial.
E	Regular
	La configuración de las funciones es fácil de realizar, sin embargo, algunas funciones no son compatibles con la última versión de Julia y además la configuración de los gráficos es más problemática.
F	Regular
	Los gráficos realizados por Julia tienen una excelente calidad de imagen y pueden ser configurables, sin embargo, este último aspecto puede ser más difícil obtener el resultado deseado.

Los resultados obtenidos en las tablas 1, 2 y 3 se concentran en un gráfico de frecuencias que se muestra en la figura 3. En este gráfico se agrupan por lenguaje cuántas valoraciones de cada tipo obtuvieron.

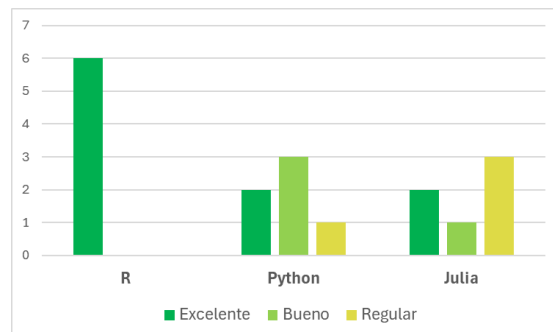


Figura 3. Gráfica de frecuencias de las evaluaciones

5. Análisis de Resultados

A partir de los resultados se puede observar que el lenguaje que más valoraciones excelentes obtuvo fue R mientras que el lenguaje que obtuvo más valoraciones regulares fue Julia. Estos datos permiten obtener información de que, de los tres lenguajes utilizados en este experimento, R es el que sería más sencillo y adecuado para utilizar como apoyo en la probabilidad y estadística.

Algo a destacar es que ninguno de los lenguajes obtuvo valoraciones en malas o muy malas, lo que significa que se encuentran dentro de rangos aceptables aun con pocas nociones de programación, además de que los tres están valorados como excelente en la complejidad del lenguaje.

Un aspecto para destacar es que, si se desea utilizar Julia como herramienta de apoyo en Probabilidad y Estadística, se debería tener el tiempo y dedicación de buscar documentación sobre las funciones que utiliza, ya que en comparación con los otros dos lenguajes su contenido en la red es muy bajo.

Mientras que Python obtuvo principalmente valoraciones buenas y excelentes, y su única valoración regular es en la instalación de plugins o módulos, por lo tanto, si no se requieren de módulos fuera de la instalación estándar del intérprete, es posible considerarlo una herramienta eficaz para cualquier persona con poco conocimiento en programación y con la necesidad de resolver problemas específicos de probabilidad y estadística.

6. Conclusiones

Los aspectos más importantes para resaltar en este artículo es que se aprovecharon los lenguajes de programación existentes para resolver problemas de probabilidad y estadística, en este caso particular todas las personas involucradas tenían conocimientos previos de programación en otros lenguajes, lo que les permitió adaptarse rápido a los lenguajes utilizados para este artículo. Por lo tanto, sería importante valorarlo con personas que tuvieran menos experiencia en programación.

Además, en este artículo se muestran valoraciones sobre la sencillez y versatilidad de los lenguajes de programación, donde R fue el que obtuvo mejores resultados. Sin embargo, no se evalúa en ningún momento cuál es más eficiente que el otro, ¿Si se evaluara eficiencia, R seguiría siendo el mejor?

Esto último es una propuesta de trabajo futuro, que permitiría llevar esta comparación a un nivel más profundo, con la implementación de una metodología diferente, enfocada a generar resultados cuantitativos para realizar una comparación equitativa.

7. Referencias

- [15]M. Spiegel, *Estadística*, España: McGraw-Hill, 2013.
- [16]D. Lind, R. D. Mason y W. G. Marchal, *Estadística para administración y economía*, España: McGraw-Hill, 2000.
- [17]G. Posada, *Elementos básicos de estadística descriptiva para el análisis de datos*, Colombia: Fundación Universitaria Luis Amigó, 2016.
- [18]D. Anderson, D. Sweeney y T. Williams, *Statistics for Business and Economics (10a. ed.)*, E.U.A.: South-Western College Pub, 2018.
- [19]Z. Ghahramani, "Probabilistic machine learning and artificial intelligence", *Nature*, Vol. 521, No. 7553, pp. 452–459, 2015, Disponible: <https://doi.org/10.1038/nature14541>
- [20]The R Foundation, *What is R*, 2025. [En línea]. Disponible: <https://www.r-project.org/>

- [21]Python Software Foundation, *El tutorial de Python*, 2025. [En línea]. Disponible: <https://docs.python.org/>
- [22]A. Downey, B. Laurens, *Introducción a la programación en Julia*, 2018. [En línea]. Disponible: <https://introajulia.org/>
- [23]R. Fisher, "Iris", UCI Machine Learning Repository, 1936. [En línea]. Disponible: <https://doi.org/10.24432/C56C76>.
- [24]K. Hamdiah, "Superconductivity Data", UCI Machine Learning Repository, 2018. [En línea]. Disponible: <https://doi.org/10.24432/C53P47>.
- [25]A. Janosi, W. Steinbrunn, M. Pfisterer, y R. Detrano, "Heart Disease", UCI Machine Learning Repository, 1989. [En línea]. Disponible: <https://doi.org/10.24432/C52P4X>.