



Revista digital

**Matemática, Educación e Internet**

(<https://tecdigital.tec.ac.cr/revistamatematica/>).

Vol 20, No 2. Marzo – Agosto, 2020

Artículo de sección

ISSN 1659 -0643

## Estadística computacional con Julia. Ventajas de su enseñanza en la universidad

Computer statistics with Julia. Advantages of your teaching at the university

**Humberto Cuevas**

jesus.ca@chihuahua2.tecnm.mx

Profesor en Ciencias Básicas

Instituto Tecnológico de

Chihuahua II/TecNM

México

**Gabriela Camacho-Ríos**

gabycamachorios@hotmail.com

Profesora en Informática

Instituto Tecnológico de

Chihuahua II/TecNM

México

**Cristina Solís**

csolismoreira@gmail.com

Consultora-Productora

Educativa

Costa Rica-México

Recibido: 18 mayo 2019

Aceptado: 1 octubre 2019

**Resumen.** En este artículo se describen las características más representativas del lenguaje de programación Julia. Se hace énfasis en presentar algunos de los paquetes más importantes para cómputo estadístico y se examina su pertinencia de incorporar el uso del lenguaje en cursos de estadística en instituciones universitarias. Se revisa la importancia de utilizar pruebas estadísticas paramétricas, no paramétricas y robustas; posteriormente, se esboza un método para incorporar su empleo en la enseñanza.

**Palabras clave:** estadística computacional, estadística robusta, educación estadística, lenguaje Julia

**Abstract.** This article describes the most representative features of the Julia programming language. The emphasis is placed on presenting some of the most important packages for statistical computation and their relevance to incorporate the use of language in statistical courses in university institutions is reviewed. The calculation of parametric, no parametric and robust statistics test are reviewed; subsequently, one method is outlined to incorporate its use in the classroom.

**KeyWords:** computational statistics, robust statistical, statistical education, Julia language

## 1.1 Introducción

---

¿Cuáles son las características más representativas del lenguaje de programación Julia? ¿Cuáles son los paquetes más relevantes para el cálculo estadístico? ¿Por qué es importante incorporar su uso en la educación universitaria? Muchos lenguajes de programación fueron creados para utilizarse en el desarrollo de aplicaciones de propósito general; posteriormente, se diseñaron paquetes especializados en áreas técnicas y científicas particulares para diversificar su uso. Otros más fueron diseñados desde el inicio con propósitos específicos y altamente especializados.

Python y R constituyen ejemplos representativos de lo anterior. Según Solano [24], el primero fue diseñado para permitir diferentes paradigmas de programación (v.gr. orientación a objetos, imperativa y funcional), sintaxis con alta legibilidad, curva de aprendizaje corta y facilidad para crear extensiones. En el caso de R, Wickham [27] señala que se creó para efectuar análisis estadístico y elaborar gráficas sofisticadas como apoyo a su enseñanza.

La realidad es que ambos lenguajes tienen una larga tradición en el entorno industrial y académico; comparten el liderazgo en muchas áreas representativas de la *Ciencia de Datos e Inteligencia Artificial*; también, cuentan con comunidades de desarrollo entusiastas, organizadas y altamente especializadas en Tecnología Computacional, Estadística, Matemática e Ingeniería. Entre las comunidades más representativas se pueden enunciar las siguientes, a saber, *Python Organization, Python Community, PyLadies, Pyslackers, R Consortium, R-bloggers, ROpenSci, R User Groups, R-Ladies*, entre otras.

Es importante destacar que un rasgo distintivo de las comunidades de apoyo constituye la promoción de la igualdad y equidad de género en el uso de estos lenguajes, lo anterior está permitiendo acercar más a las mujeres al estudio de profesiones relacionadas con la Ciencia, Tecnología, Ingeniería y Matemáticas (STEM por sus siglas en inglés) .

No obstante, el desarrollo de ambos lenguajes y el crecimiento constante de sus comunidades que las impulsan, han motivado el surgimiento de otros lenguajes de programación que intentan aprovechar, de mejor manera, los avances en el desarrollo del hardware, hacer más eficiente el cómputo paralelo, reducir la curva de aprendizaje y concatenar librerías de Python y R dentro de su código.

El objetivo de este artículo es describir los atributos más significativos del lenguaje de programación Julia, presentar algunos de los paquetes más importantes para el cálculo de estadísticos robustos y examinar la pertinencia de su utilización en el ámbito universitario, especialmente en las áreas de Ciencias Básicas e Ingeniería. De igual manera, se delinea un método para incorporar su empleo en la enseñanza de temas estadísticos elementales.

## 1.2 Antecedentes del lenguaje de programación Julia

---

Julia se originó por la necesidad de contar con un lenguaje moderno, rápido, legible y multiplataforma para su uso en aplicaciones técnicas, académicas y científicas. Respecto de las disciplinas beneficiadas de manera directa, se pueden enunciar las siguientes: cálculo infinitesimal, ecuaciones diferenciales,

álgebra matricial, geometría, investigación de operaciones, estadística, física cuántica, astrofísica, bioquímica, bioinformática, genética, ecología, criptografía, economía, finanzas, entre otras. Sus inicios se remontan al año 2009, con los trabajos de Jeff Bezanson, Stefan Karpisnki, Viral B. Shah y Alan Edelman [3], pero, es en 2012 cuando crearon un sitio web alusivo al proyecto de desarrollo; lo cual ha permitido la conformación de una comunidad de usuarios que contribuyen de manera entusiasta y comprometida a su desarrollo. Los creadores concibieron el proyecto de trabajo estableciendo parámetros comparables con otros lenguajes de programación; asimismo, se planeó que su curva de aprendizaje fuese similar a la de Python, que incorporara el dinamismo de Ruby, la velocidad de C, la capacidad de procesamiento cadena de caracteres de Perl, la facilidad para cálculo matricial de Matlab; además, la potencia en análisis y visualización de datos en R.

En un estudio conducido por Popuri [21], que tuvo como propósito medir y comparar el desempeño entre Matlab, Octave, R y una versión antigua (v. 0.5) de Julia. Utilizaron un equipo con 72 nodos, cada uno con dos procesadores Intel E5-20650v2 Ivy Bridge de 2.6 GHz, 20 MB de memoria caché y ocho núcleos por cada unidad central de procesamiento, es decir, 16 núcleos por cada nodo. Dichos nodos tenían 64 MB de memoria de acceso aleatorio. Las pruebas utilizadas para las mediciones consistieron en solución de ecuaciones lineales del siguiente tipo:

$$\begin{aligned} -x_2 + x_3 &= 3 \\ x_1 - x_2 - x_3 &= 0 \\ -x_1 - x_3 &= -3 \end{aligned} \tag{1.1}$$

También, se realizaron pruebas para cálculo numérico de eigenvalores y eigenvectores, construcción de gráficos en dos dimensiones a través del trazado de una función del tipo  $f(x) = x \sin(x_2)$ , durante el intervalo de  $[-2\pi, 2\pi]$  y una base de datos (matlabdata.dat) suministrada por una institución universitaria de investigación pública. Los autores encontraron que un factor en contra de R fue la diferencia de su sintaxis; Octave y Matlab fueron capaces de resolver problemas de gran tamaño (datos), pero Julia y Matlab estuvieron más cerca en los tiempos de ejecución absolutos. Los resultados son importantes si se considera que el desarrollo de Julia aún era incipiente.

En un estudio más reciente efectuado en la República Federal de Alemania, Weibezahn y Kendziorski [26] utilizaron Julia para un proyecto de modelado algebraico de sistemas eléctricos a gran escala en el sector eléctrico. En su informe los autores presentan y destacan las ventajas del paquete *JuMP-jl* y sus capacidades para el modelado. También, describen y comparan las características más significativas de software especializado como *GAMS* y *ECOS* con paquetes de Python y R. Enfatizan la potencia de cálculo, capacidad gráfica, el rápido desarrollo del lenguaje y la posibilidad de dar mantenimiento al código.

### 1.3 Características más representativas

---

La versión 1.0 fue liberada en agosto de 2018. Entre los atributos más representativos destaca ser de código abierto, la disponibilidad de un compilador en tiempo de ejecución, orientado a objetos, capacidad superior a versiones anteriores para efectuar cálculo en paralelo y distribuido, integración de una terminal interactiva (REPL), posibilidad de realizar llamadas a otros lenguajes como C, Python y R, así como disponer de un sitio web que permite acceder a más de 2100 paquetes especializados en varias

categorías.

La interacción con el lenguaje se puede realizar a través de diversas extensiones y plataformas de trabajo, a saber, *Jupyter Notebook*, *Codingground*, *Visual Studio Code*, *Atom–Juno*, *emacs*, *Vim*, entre otras. Es probable que pronto sea posible utilizar Julia en las plataformas tecnológicas que ofrecen servicios de almacenamiento y procesamiento de datos en la nube como Amazon Web Service (AWS), Microsoft Azure o Google Cloud.

Con Julia es posible crear informes de investigación reproducible en términos computacionales. Así, puede utilizarse el sistema de composición de textos *L<sup>A</sup>T<sub>E</sub>X* y el lenguaje de marcado *Markdown* para su elaboración, mientras se escribe código para efectuar cálculo numérico y crear representaciones gráficas sofisticadas. Esta característica coloca a Julia a la par de otros lenguajes como Python, R y Scala que, comúnmente, usan diversas comunidades epistémicas para elaborar sus reportes de investigación e impulsar la transparencia y honestidad intelectual en su quehacer científico.

También, es posible manejar datos de distinto tipo y sus correspondientes subtipos; entre los más útiles se pueden enunciar *String*, *BigInt*, *BigFloat*, *int64*, *Rational*, entre otros. El tipo *String* es útil en la manipulación de cadenas de texto; *Char* por su parte está indicado para caracteres específicos. Según De Haro [4], existe una gran cantidad de tipos de datos que pueden observarse introduciendo en la consola la instrucción `print(typeof(Any))`.

Julia permite definir fácilmente variables, construir y manejar variables, manipular vectores y generar gráficas versátiles y sofisticadas. En el Cuadro 1.1 se presentan expresiones básicas y los resultados esperados.

**Tabla 1.1:** Sintaxis de expresiones básicas con Julia y los resultados esperados.

Sintaxis	Resultado
<code>x = 5</code>	Define una variable x con un valor de 5
<code>y = [9 3 8 2 5 6 7 1]</code>	Matriz fila asignada a y
<code>B = [9 3 8 2 ; 5 5 6 7 ; 1 4 1 2 ; 7 5 3 6]</code>	Asignación a B de una matriz de 4 x 4
<code>sqrt(-2 + 0im)</code>	$\sqrt{-2}$ como un número complejo
<code>besselj(1,3)</code>	Calcular una función Bessel $J_1(3)$
<code>cos(2pi/6)</code>	Calcular el coseno de $(2\pi/6)$
<code>exp(18)</code>	$e^{18}$
<code>using Pyplot; plot(x,y)</code>	Gráfica de puntos de x y y a través del paquete Pyplot
<code>χ + → (Tabulador)</code>	Inserta el símbolo de ji-cuadrada en <i>L<sup>A</sup>T<sub>E</sub>X</i>

Al igual que Python y R, Julia integra un sistema flexible para la administración de paquetes a través de instrucciones intuitivas. En el Cuadro 1.2 se muestran dichas instrucciones y sus funciones.

**Tabla 1.2:** Administración de paquetes e instrucciones básicas.

Instrucción	Resultado
Pkg.add("paquete")	Instala un paquete externo.
using Package	Cargar un paquete instalado.
Pkg.installed( )	Lista con los paquetes instalados.
Pkg.status( )	Lista con los paquetes instalados y su versión.
Pkg.update( )	Actualizar todos los paquetes.
Pkg.rm("paquete" )	Elimina del sistema un paquete instalado con antelación.
Pkg.build( )	Inicia el proceso de reconstrucción de un paquete específico.

Las características robustas del lenguaje están permitiendo desarrollar aplicaciones de interés industrial, como la realidad aumentada, el Internet de las cosas, la impresión en 3D, la conducción automatizada de vehículos y la medicina predictiva. Dichas aplicaciones exigen uso intensivo de cómputo para el manejo de macrodatos, modelos matemáticos y estadísticos avanzados. Es probable que por esos motivos, corporaciones multinacionales como Amazon, IBM, Microsoft, Path BioAnalytics, Augmedics, entre otras, están utilizando Julia. En la Figura 1.1 se presenta el código para simular el algoritmo de ordenamiento (burbuja); nótese la flexibilidad del código y la similitud con el lenguaje de programación Python.

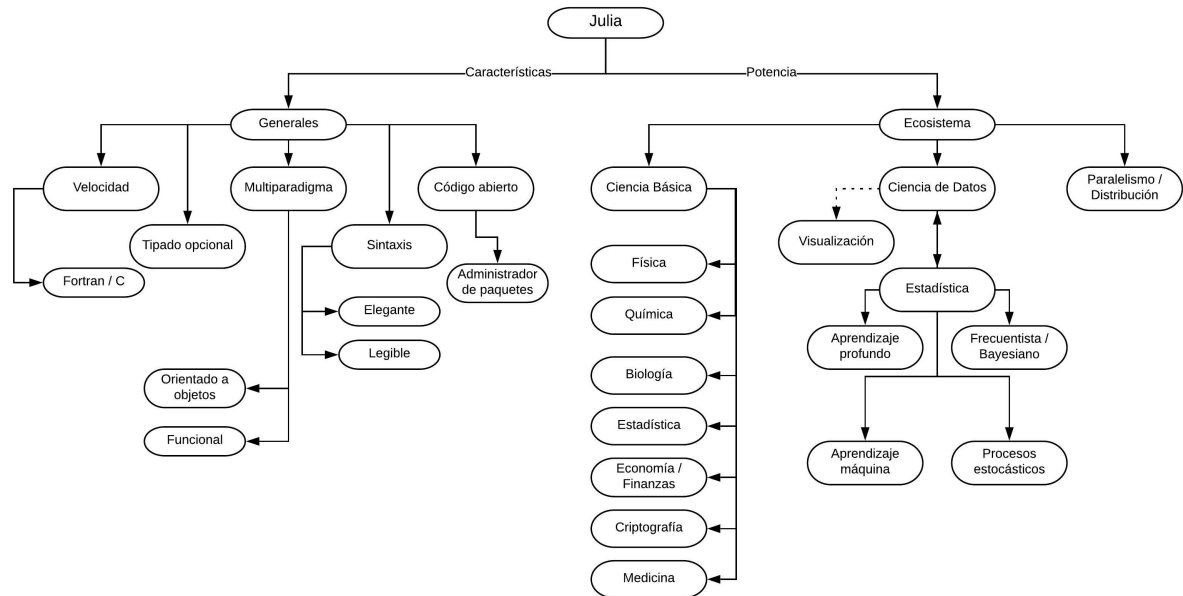
```

1  function bubbleSort!(a)
2      n = length(a)
3      for i in 1:n-1
4          for j in 1:n-i
5              if a[j] > a[j+1]
6                  a[j], a[j+1] = a[j+1], a[j]
7              end
8          end
9      end
10     return a
11 end

```

**Figura 1.1:** Código para el algoritmo de la burbuja

La diversidad de aplicaciones permitidas por el lenguaje y el apoyo de este tipo de corporaciones ponen en tela de juicio la limitación aparente de su uso exclusivo en círculos científicos y académicos. En la Figura 1.2 se presentan las características más significativas del lenguaje y algunas de las disciplinas en las cuales, actualmente, se utiliza de forma intensiva.



**Figura 1.2:** Características representativas del lenguaje Julia y sus aplicaciones.

A diferencia de otros lenguajes, en Julia no es necesario declarar cada variable que se pretenda usar. Sin embargo, para aprovechar su potencia y velocidad es recomendable efectuar este proceso y evitar errores en el proceso de ejecución. Esta es una de sus características más distintivas.

Aunque reúne características tan versátiles, existen limitaciones respecto de otros lenguajes. Las extensiones para su uso en editores de código aún están lejos de ofrecer las herramientas para aprovechar todo su potencial. Caso contrario a Python y R que cuentan con entornos de desarrollo integrado elaborados ex profeso, como Pycharm o Spyder para el primero o RStudio para el segundo. De igual forma, al ser un lenguaje nuevo, la comunidad de usuarios y desarrolladores es pequeña en cantidad, comparándola con los descritos líneas atrás.

Para ahondar más al respecto, se sugiere analizar el texto de introducción al lenguaje elaborado por De Haro [4]. El autor expone, de manera clara, las capacidades del lenguaje, la administración de sus paquetes y presenta una variedad importante de ejemplos. De igual forma, contribuyen los primeros capítulos del trabajo de Sandeep Nagar [19]. El texto de Zacharias Voulgaris [25] es una opción valiosa en el campo de la Ciencia de Datos porque trata tópicos relacionados con el análisis exploratorio y confirmatorio, modelos de regresión, redes neuronales, árboles de decisión e incluye un apéndice de cómputo paralelo.

## 1.4 Estadística computacional

Resulta complejo establecer una definición precisa del término estadística computacional. En una primera aproximación, se puede definir como la disciplina que examina los efectos de los desarrollos de la computación en la estadística, es decir, el impacto en el análisis exploratorio y descriptivo de

datos, estadística robusta, métodos de simulación, aprendizaje profundo, inferencia, análisis bayesiano, cálculo recursivo, entre algunos cambios más.

No obstante, la primera aproximación pareciera que condiciona la evolución de la estadística al desarrollo computacional. Otra aproximación, más justa, puede ser la que propone Sawtzki [23] al mencionar que la estadística computacional hace alusión a los desarrollos metodológicos realizados, tanto en el campo de la estadística como de la computación; pero, explica la existencia de una influencia recíproca. El autor también destaca que al compararse con la acepción tradicional de la estadística, la diferencia más notable estriba en el impacto del cómputo de la formulación, perfeccionamiento, implantación, adaptación y monitoreo de avances matemáticos, como sucede con los algoritmos.

Por otra parte, la *International Association for Statistical Computing* (IASC) [17], es una de las asociaciones reconocidas por el *International Statistical Institute* (ISI), máximo rector internacional en el campo. En los estatutos de la IASC, promulgados en el mes de enero del año 2015, se subraya la importancia de promover la teoría, métodos y práctica de la estadística computacional y ser un puente entre investigadores y profesionales de la estadística, la computación y otras disciplinas científicas.

En este artículo, se asume que la estadística computacional tiene un espacio propio que transita entre dos áreas de la ciencia claramente diferenciadas, pero, complementarias entre sí. La estadística integra métodos para el acopio apropiado, organización, uso óptimo e interpretación de la información extraída de los datos para explicar fenómenos complejos, formular explicaciones y generar conocimiento. La computación, por su parte, auxilia a la estadística en el procesamiento y representación \_ numérica, gráfica y tabular\_ de información de manera rápida, legible y extrapolable. Así, en este documento se adopta la tesis de la IASC, la cual presenta a la estadística y a su praxis como disciplina de unión y no de división con otros campos de la ciencia, como la computación. Todo en aras de buscar el desarrollo científico e impulsar la solución de problemas sociales de largo alcance, sin distinciones de índole geográfica, económica, política o credo religioso.

Como se indicó líneas atrás, Julia es un lenguaje versátil, rápido, potente y legible que permite utilizarse en entornos técnicos, académicos, científicos y en el desarrollo de aplicaciones de propósito general. En el caso particular del ámbito técnico-académico-científico, sobresalen sus atributos para trabajar con *valores ausentes*, tanto en su uso como en representación visual. Así, en cualquier colección de datos es posible permitir elementos sustitutos. Por tanto, las comunidades de trabajo que cultivan disciplinas relacionadas con la probabilidad, la estadística y el análisis de datos se benefician de la flexibilidad y velocidad de esta característica y del compilador subyacente.

### Paquetes representativos para el análisis estadístico

En virtud de que es un lenguaje joven, la cantidad de paquetes creados para el cómputo estadístico es apenas una fracción de los existentes en Python y, significativamente, menores a los desarrollados para R. Sin embargo, los paquetes existentes integran herramientas para efectuar cálculos complejos en campos destacados de la Estadística. Algunos de los más representativos se describen a continuación, de forma breve.

**StatsBase.** Este paquete ofrece las funciones básicas para efectuar resúmenes estadísticos; muestreo \_con reemplazo y sin él\_, estimación, matrices de covarianzas, análisis de correlación, cálculos robustos básicos, entre otras.



**StatsModels.** Con este paquete es posible realizar ajuste y evaluación de modelos estadísticos. Una característica fundamental es que proporciona una Interfaz de Programación de Aplicaciones (API) para el desarrollo de otros paquetes o potenciar los existentes. Los Modelos Generales Lineales (GLM) son un ejemplo de lo anterior; también, las series de tiempo y la posibilidad de tabular datos.

**DataFrames.** Permite construir arreglos de datos, su manejo y ordenamiento en función de las necesidades de cómputo. Sobresale la capacidad de trabajar con vectores y enfrentar el reto que constituyen los valores ausentes, muy comunes en las bases de datos. Es probable que debido al uso de datos masivos (macrodatos) y su organización poco convencional, este paquete sea de los más utilizados y con desarrollo más acelerado.

**Distributions.** Integra una colección de distribuciones de probabilidad univariadas, multivariadas, con arreglos, modelos mixtos, así como ajuste de distribuciones (v.gr. máxima verosimilitud y a posteriori).

**HypothesisTests.** Incluye métodos para efectuar estimación por intervalos; pruebas paramétricas, no paramétricas como las de Anderson-Darling, Binomial, F, Kolmogorov-Smirnov, Kruskal-Wallis, Wilcoxon, Mann-Whitney, Signos, Permutaciones, además, series de tiempo como, Durbin-Watson, Breusch-Godofredo, Jarque-Bera, Durbin-Watson, Box-Pierce y Ljung-Box, entre otras.

**MultivariateStats.** La capacidad para realizar Análisis Multivariante es posible a través de uno de los paquetes más versátiles y mejor desarrollados hasta el momento. Es posible efectuar análisis de regresión, factorial, de componentes principales, de correlación canónica, discriminante y coadyuvar en el proceso de *limpieza* de datos. Destaca el análisis de componentes independientes, por ser una técnica de índole computacional que permite separar señales multivariadas en otros subcomponentes que no son gaussianos, pero, sí independientes.

**TimeSeries.** Este paquete incorpora herramientas para representar, manipular y aplicar rutinas de cómputo a series de tiempo; permite usar operadores matemáticos, lógicos y de comparación. Además, es posible elaborar gráficas en conjunción con el paquete *Plots.jl* que debe ser instalado por separado, pero que se encuentra en continuo desarrollo.

**GLM.** Con StatModels es posible realizar evaluaciones de GLM, pero existe un paquete diseñado especialmente para potenciar su análisis. Una característica importante es que los nombres de las funciones son similares a las del lenguaje R. Lo anterior coadyuva en reducir la curva de aprendizaje, y así, aprovechar que dichas funciones se han probado de manera exhaustiva desde hace dos décadas.

**Klara.** Con este paquete es posible trabajar Métodos de Montecarlo utilizando Cadenas Markov (MCMC). Por tanto, las comunidades epistémicas que cultivan el tratamiento de casos relacionados con juegos de azar, meteorología, epidemiología, simulación de procesos estocásticos, análisis econométricos, cálculo de volúmenes complejos; se benefician de forma directa. El algoritmo Metropolis-Hastings es uno de los más comunes dentro de los que incorpora el MCMC y muy fácil de implementar en Julia:

$$\alpha(x_{t-1}, x') = \min \left( 1, \frac{p(x')\pi(x_{t-1} | x')}{p(x_{t-1})\pi(x' | x_{t-1})} \right) \quad (1.2)$$

Una opción rápida en el aprovechamiento de los paquetes esenciales para el cómputo estadístico es el meta-paquete *StatsKit*. En los reportes de Flach [6] y Batisde [2] puede observarse el uso de Julia en procesos de cómputo exhaustivo y generación de representaciones visuales complejas. La Figura 1.3



muestra algunos de los paquetes descritos antes y otros más, los cuales permiten el modelado de áreas que intersectan la Estadística con disciplinas robustas y articuladas como la Inteligencia Artificial.

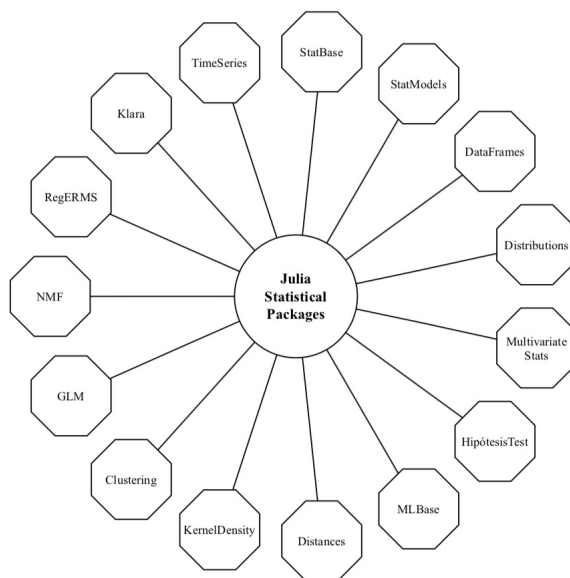


Figura 1.3: Julia Statistical Packages

### Apuntes generales

La estadística computacional está permitiendo atender problemas de distinta índole y explicar fenómenos multidisciplinarios. De igual manera, está coadyuvando en la exploración desde otras perspectivas en otros campos de la ciencia básica y aplicada (v.gr. Física de Altas Energías, Ingeniería Genética, Biofísica, Climatología, ...). El tratamiento y análisis de datos masivos en estos campos requiere contar con equipo de cómputo e informático sofisticado; es común encontrar valores anómalos y extremos que implica el uso de métodos estadísticos robustos y procesos avanzados de modelado.

El lenguaje de programación Julia es una alternativa adecuada debido a su potencia de cálculo; curva de aprendizaje poco pronunciada; capacidad de interactuar con paquetes de lenguajes con una larga tradición en el ámbito científico como Python, R y C; comunidad de usuarios en continuo crecimiento y su código abierto que facilita el desarrollo de sus capacidades y adaptación a objetivos específicos. Además, muchas de sus funciones más importantes constituyen una evolución de paquetes de R (v.gr. WRS, WRS2, RDatasets,...), pero aprovechando la arquitectura más moderna de Julia.

## 1.5 Estadística computacional en el ámbito universitario

Es necesario valorar la conveniencia de incorporar un lenguaje de programación que pueda ser utilizado para el cómputo científico y el desarrollo de aplicaciones en la educación universitaria, especialmente en la enseñanza de la estadística por su rol metodológico, herramental y disciplinar.

Actualmente, Julia se usa en actividades de docencia e investigación, principalmente en universidades de reconocido prestigio de América, Europa y Oceanía; entre las más destacadas se pueden señalar al Instituto Tecnológico de Massachussets, las Universidades de Harvard, Brown, Berkeley, Colorado,

Londres y la Nacional de Australia. En América Latina destacan universitarias de Argentina, Brasil, Colombia, México y Perú.

Ciertamente, el uso principal de Julia en el ámbito universitario se ha circunscrito al análisis matemático, estadístico y visualización de datos en estudios de posgrado, especialmente en programas de posgrado. Afortunadamente, el apoyo masivo que está recibiendo el lenguaje, especialmente por comunidades externas a la academia, impulsa su utilización en niveles de pregrado. Es probable que la versatilidad, potencia y capacidad de crear paquetes permitan que Julia transite, rápidamente, de ser una opción de estudio a un integrante central en planes y programas académicos.

La literatura disponible para su uso en la enseñanza es todavía incipiente. No obstante, comunidades de expertos están elaborando material de trabajo y poniéndolo a disposición de investigadores, profesores y estudiantes para su estudio. Destacan Sargent y Stachurski [22] quienes ofrecen un conjunto de manuales técnicos relacionados con programación computacional, matemáticas, estadística y su aplicación a la economía y análisis financiero, a través del uso de Python y Julia. Sugieren el uso de Python a los usuarios noveles debido a que tienen una comunidad de usuarios mayor que Julia; no obstante, advierten que este último tiene características más avanzadas y su evolución es muy rápida entre otros factores porque muchas de sus librerías se escriben en el mismo lenguaje. Los autores, también, destacan el potencial de otros lenguajes como R, C, C++, Fortran y Matlab, aunque se decantan por Python para los principiantes y Julia para usuarios especializados.

Python y R son líderes indiscutibles en el análisis estadístico, numérico, visualización de datos y aprendizaje automático. El primero es fácil de aprender, tiene miles de paquetes para realizar prácticamente cualquier análisis; es de propósito general y tiene una larga tradición en las comunidades provenientes de la informática y ciencia computacional. El segundo proviene del entorno de la estadística, la probabilidad y el análisis de datos; muchas de las funciones más potentes se encuentran en el núcleo, mientras que otros lenguajes deben importar librerías para su acceso. También, cuentan con entornos de desarrollo integrado creados ex profeso. Sin embargo, ambos tienen limitaciones, una de ellas es que requieren vectorizar funciones para tener un desempeño aceptable aún cuando muchas librerías están escritas en C. Julia por su parte, es una alternativa asequible, potente y versátil para su incorporación en los planes y programas de estudio universitarios, especialmente en aquellos que impliquen el uso de datos masivos, simulación computacional, detección de patrones, visualización avanzada, procesos estocásticos, modelado matemático, entre otras actividades de misión crítica.

### Métodos estadísticos paramétricos, no paramétricos y robustos en la enseñanza

Habitualmente, en los programas académicos se contempla la enseñanza de pruebas paramétricas y no paramétricas. En el caso de las primeras, se exige para su aplicación al menos el cumplimiento de supuestos específicos como *normalidad* y *homocedasticidad*; la prueba *t-Student* y el Análisis de Varianza (ANOVA) son ejemplos clásicos. Respecto de las segundas, no contemplan el cumplimiento estricto de los supuestos, por tanto, se les conoce como pruebas libres de distribución; las pruebas de bondad de ajuste *Chi-cuadrado* para variables categóricas, *Binomial* para proporciones, *Kolmogorov-Smirnov* para variables cuantitativas, o las pruebas comparativas como las de *U de Mann-Whitney*, *Fisher*, *Wilcoxon*, *McNemar*, *Kruskal-Wallis*, *Friedman* o *Q de Cochran* son algunas de las más representativas. La potencia de las pruebas no paramétricas es menor respecto de las clásicas, pero, su utilización es útil cuando los tamaños de muestra son pequeños ( $n < 30$ ), porque no son afectadas en la misma magnitud por no distribuirse normalmente y no cumplir con el supuesto de homogeneidad de varianza.

Así, la aplicación de los métodos clásicos estadísticos implica el cumplimiento de condiciones muy rígidas. Según García [20], entre estas condiciones se encuentra el cumplimiento de un modelo de distribución fijo con indeterminación de uno o dos parámetros, típicamente la  $\mu$  y  $\sigma^2$ . También, señala la dificultad de que los modelos probabilísticos se ajusten al fenómeno aleatorio observado y advierte del riesgo de perder validez en los resultados que se obtengan.

Ferrero [5] menciona que es frecuente encontrar conjuntos de datos con valores extremos (outliers) cuya consecuencia inmediata se refleja en las mediciones estadísticas y la disminución de la capacidad para localizar efectos reales. Sin embargo, la autora no recomienda utilizar métodos robustos cuando los datos son altamente asimétricos o tienen una distribución multimodal, es decir, estos métodos son útiles por su potencia, pero de ninguna manera constituyen una panacea.

Los principios modernos de los métodos robustos se pueden encontrar en los trabajos de Peter Huber [10], [11], [12], [13], [14], [15], [16]. Quizá la mayor aportación de este autor fue el fundamento matemático y en consecuencia la aceptación por parte de la comunidad académica internacional.

Frank Hampel, también colaboró en el desarrollo teórico iniciado por Huber, además logró definir *robustez* en términos estadísticos y la *curva de influencia*. Sus contribuciones más representativas se pueden encontrar en [7], [8] y [9].

Otro investigador destacado en el campo de los métodos robustos es Rand Wilcox. Entre sus aportaciones se encuentra la creación del paquete *WRS* cuya versión fue presentada como parte de uno de sus libros [28]. El paquete fue diseñado para el lenguaje R, pero muchas de sus funciones se han adaptado para utilizarse con Julia.

Los métodos robustos son poco utilizados, quizá por desconocimiento o por el paradigma dominante que promueve los métodos clásicos en muchos ámbitos. Es común escuchar, en pasillos universitarios y de centros de investigación, que las pruebas paramétricas son resistentes a las variaciones en los supuestos de normalidad y homogeneidad de la varianza (homocedasticidad); sin embargo, ese argumento puede ser válido únicamente cuando se examina el efecto de variaciones pequeñas, no en desviaciones grandes como es común encontrar en los datos reales.

Existen paquetes desarrollados para el uso de métodos robustos como *RobustStats*. Este paquete integra funciones útiles agrupadas con base en estimadores de ubicación, dispersión, cálculo de intervalos de confianza, así como otras funciones de utilidad. La mayoría se apoyan en el método de *bootstrap*. En el caso de los estimadores de ubicación, es posible calcular las medias acotada y winsorizada, la medida de ubicación  $\tau$  de Yohai y Zamar, el estimador paso a paso a través del método  $\psi$  de Huber, entre otras. Entre los estimadores de dispersión se encuentran la varianza y covarianza winsorizada, la curva de varianza media (medida en porcentaje), la escala  $\tau$  de Yohai y Zamar, el rango intercuartil normalizado y el estadístico de Rousseeuw y Croux. Respecto del cálculo de intervalos de confianza y error estándar, se pueden mencionar la detección de valores anómalos (atípicos) y la función  $\psi$  de Huber. En el sitio <https://github.com/mrxiaohe/RobustStats.jl> es posible encontrar un listado completo de las funciones incluidas en el paquete, el código necesario para su uso y fuentes de información. No obstante, se sugiere contactar al equipo que impulsa su desarrollo porque las funciones descritas antes, únicamente, funcionan con versiones beta de Julia.

En el ámbito universitario (pregrado y posgrado), la inclusión de métodos robustos en los programas de estudio es reducida, al menos en el nivel de pregrado. En posgrado es posible cursar algunos temas incluidos en materias que privilegian los métodos clásicos; una excepción constituyen los programas específicos de estadística y actuaria, que reconocen las ventajas de su uso.

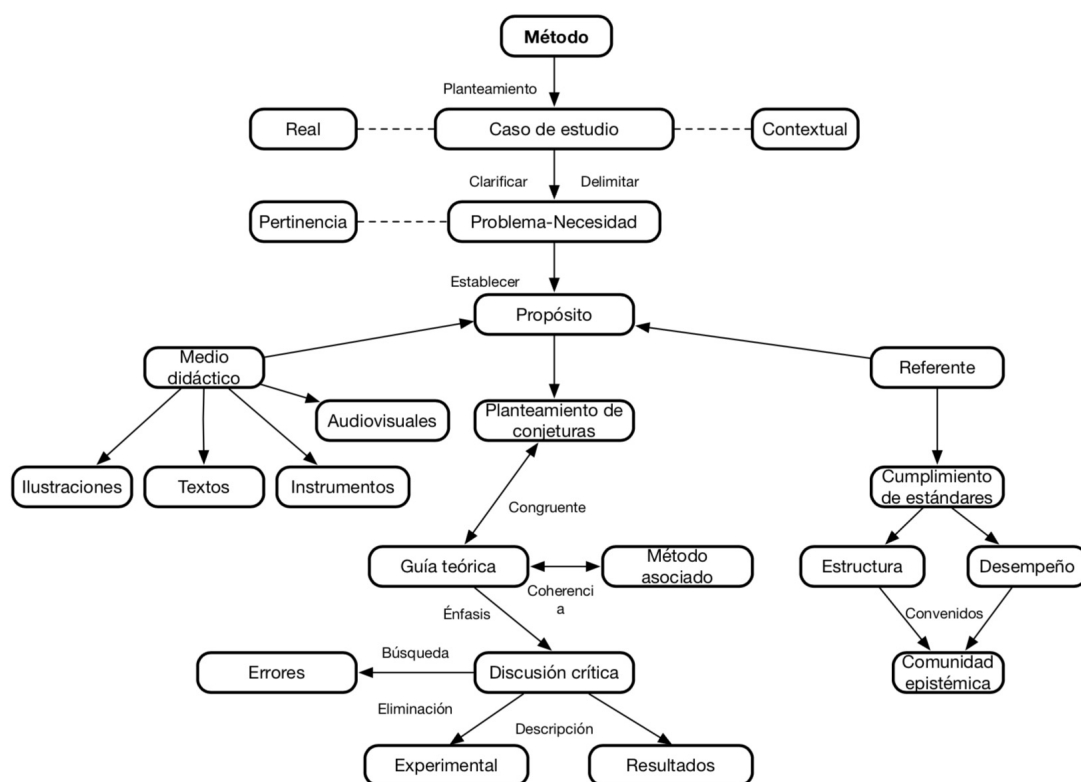
En este artículo, se parte de la hipótesis de sugerir la necesidad de que el profesorado coadyuve para que el alumnado sea más riguroso en el uso de los métodos en función del caso por tratar; y así, efectuar demostraciones para notar las diferencias en los resultados obtenidos en la aplicación de pruebas de distinto tipo y, además, verificar que en el caso de los métodos clásicos, únicamente, son resistentes en casos limitados.

#### Método para incorporar la estadística computacional con Julia

El método que se propone para tratar tópicos estadísticos con el auxilio de la programación computacional tiene como base el enfoque teórico y metodológico del Aprendizaje Basado en Desafíos. En este enfoque se parte de la tesis que señala la necesidad de que los estudiantes participen de manera activa en proyectos de trabajo poco o nada estructurados. Como señala Akella [?], el aprendizaje se vive y permite combinar el comportamiento con la cognición.

El Aprendizaje Basado en Desafíos tiene sus orígenes en los trabajos de intelectuales de la talla de John Dewey y David Kolb; en el caso de este último, su modelo describe al aprendizaje como el resultado integral de la forma en la que las personas perciben y procesan una experiencia [18]. En este enfoque se concibe al aprendizaje y al conocimiento como una acción, el *saber hacer* o *actuar* de un aprendiz respecto de un tema o problema de estudio le permite *conocer*. Por tanto, ubica a los aprendices en el centro del proceso educativo y al profesorado como eje toral en el logro de los objetivos propuestos. En el método que se propone a continuación, al *saber hacer* se adiciona el *saber usar* y *saber comunicar* de forma articulada.

El aprendizaje basado en retos, también, comparte atributos con otros enfoques como el Aprendizaje Basado en Proyectos y en el Aprendizaje Basado en Problemas, especialmente en lo relacionado a la participación activa en casos de estudio reales y contextuales. En la Figura 1.4 se muestra la estructura del método por seguir para tratar temas estadísticos con estudiantes universitarios. Es importante subrayar que dicho método se ha aplicado con fines meramente exploratorios en dos ocasiones; la primera de ellas, con un grupo de 10 estudiantes de pregrado (Ingeniería) y, la segunda, con tres estudiantes de posgrado (Maestría).



**Figura 1.4:** Método propuesto para aprender a través de desaf

Inicialmente, es necesario plantear un caso de estudio que implique el acopio de datos reales y contextuales. Es recomendable que dicho caso sea generado de manera grupal junto al profesor porque esa es, precisamente, una de las diferencias del Aprendizaje Basado en Desafíos respecto de los enfoques mencionados líneas atrás (proyectos y solución de problemas).

Se propone conducir una o dos sesiones de trabajo en la que se hagan propuestas y el profesor coadyuve en el proceso participando como moderador. Se sugiere que las temáticas reúnan ciertas características como: [1] relación directa al programa académico que cursan los participantes; [2] examen de un problema o fenómeno social que sea percibido como tal en la sociedad o autoridades gubernamentales; [3] aunado a la característica anterior, es fundamental que la temática se encuentre entre las áreas prioritarias de la nación (v.gr. Agua, Agricultura, Alimentos, Salud, Comunicaciones, Energía, Medio Ambiente y Transporte) a la que pertenezcan los aprendices, o al menos una necesidad particular en su comunidad; [4] factibilidad y viabilidad son dos características elementales, el grado de dificultad y la inversión inherente en recursos debe ser acorde al grado de competencia de los participantes; [5] es indispensable que los participantes se interesen en su desarrollo.

La función del profesorado coadyuva en clarificar, delimitar el caso de estudio, es decir, determinar si es un problema o una necesidad; posteriormente, examinar su pertinencia a través de un intercambio de ideas con argumentos sólidos. Una revisión exhaustiva de la literatura sobre el tema es imperativa, únicamente, de esa manera se podrá emitir argumentos a favor o en contra de las propuestas y, en su momento, acotar el caso de estudio. De ahí es posible derivar la justificación de su realización.

Las actividades anteriores son quizá las más complejas. Si el caso por examinar es claro, está delimitado, es factible, viable y pertinente, es posible derivar un propósito general que esté en consonancia con lo expuesto antes. Debe redactarse en un sólo párrafo, de ser posible con un máximo de tres enunciados.

Enseguida, es necesario plantear conjeturas o supuestos. En el caso de las primeras, deben ser susceptibles de ser probadas empíricamente, medibles o contables, además, deben ser definidas con claridad a la luz del planteamiento del caso, de los objetivos y la revisión previa de la literatura.

Es indispensable contar con una guía teórica y metodológica congruente con lo antes expuesto. La elección de dicha guía se hará en función de su pertinencia y no a su popularidad; el profesorado debe apoyar a los aprendices en la selección más apropiada y haciendo hincapié en la promoción de una discusión crítica en que la búsqueda, aprovechamiento y eliminación de errores tenga un papel primordial. La discusión, también, debe contemplar el diseño experimental o cuasiexperimental \_ si es necesario\_ y los medios para informar de los resultados que se obtengan.

Se aconseja el uso de medios didácticos diversos (v.gr. ilustraciones, textos, audio, video, instrumentos de medición,...) por parte del profesorado y sus aprendices. El uso de cada uno debe tener su justificación pertinente y especificar, en el caso de los instrumentos de medición, sus tolerancias y márgenes de error.

Actualmente, vivimos en una sociedad globalizada en la que existen estándares que deben acatarse de forma irrestricta, especialmente en el ámbito profesional. La institución escolar es un espacio adecuado para dar a conocer a los futuros profesionales, respecto de los estándares de cumplimiento mínimo que deben alcanzar. Es imperativo que el profesorado presente dichos estándares; deben describir parámetros de medición en términos de desempeño y estructura, comúnmente aceptadas por las comunidades epistémicas más representativas en su área de especialidad. En virtud de la importancia de esta fase, se recomienda al profesorado que muestre a sus aprendices un *objeto referente*, que puede ser un reporte de investigación, un prototipo, o bien, un producto que considere conveniente.

En el Anexo A, se presenta un caso de estudio que implica el uso de métodos estadísticos clásicos y robustos que puede servir como guía. La descripción se hace de manera sucinta e interesa que los estudiantes participen en el diseño del estudio, acopio de datos, aplicación de métodos estadísticos y, además, usen el lenguaje de programación Julia para el proceso de cálculo, representación gráfica e interpretación de los resultados, a la luz de los estándares internacionales propuestos por la Organización Mundial de Salud. En los Anexo 2.1 y 2.2 se presenta un código de ejemplo que permite analizar algunas variables. Es necesario aclarar que no todos los paquetes funcionan con la versión más actual de Julia (v. 1.2). En el caso del paquete *RobustStats*, es necesario ejecutar sus funciones en una versión anterior, especialmente la v. 0.5.

## 1.6 Conclusiones

---

El lenguaje de programación Julia nació con el propósito de realizar procesos de cómputo sofisticados como cálculo en paralelo y distribuido. Es de código abierto, dispone de un compilador en tiempo de



ejecución, permite programar con orientación a objetos y su sintaxis se asemeja al lenguaje natural.

Paulatinamente, ha ganado espacio en centros de investigación y en el ámbito universitario. Corporaciones multinacionales usan el lenguaje para modelado matemático, simulación computacional, ciencia de datos y diseño de aplicaciones de misión crítica. La comunidad de usuarios y desarrolladores se incrementa de forma constante y comienzan a multiplicarse los foros científicos de difusión \_ verbales y escritos\_ en el plano internacional.

Los lenguajes Python y R son los líderes actuales en las áreas que Julia empieza su arribo. El catálogo de paquetes especializados no es comparable; Julia está muy rezagado aún; sin embargo, el alto grado de adopción de este último en los últimos cinco años permite avizorar un futuro muy promisorio, especialmente, porque las comunidades de desarrollo se traslapan.

La estadística computacional se beneficia del desarrollo de Julia porque junto a Python y R conforman una triada que permite tratar fenómenos que hace una década sería casi imposible abordar. El llamado de funciones de un lenguaje a otro posibilita generar una sinergia entre los tres, es por esta razón que en este artículo se considera importante incorporar su estudio en la enseñanza de la estadística.

Al igual que se crean lenguajes de programación nuevos con capacidades superiores a otros de antaño, también la estadística como disciplina continúa su evolución. Así, los métodos paramétricos (clásicos) y los no paramétricos dominan el espectro de los planes de estudio, no obstante, la cantidad masiva de datos y sus características hacen difícil que se cumplan los supuestos irreductibles para aplicar los métodos paramétricos y la potencia de los no paramétricos generan dudas por la reducción de su potencia. Los métodos robustos constituyen una alternativa factible y viable porque no son tan sensibles a los valores anómalos y a la ausencia \_ alta\_ de normalidad y homocedasticidad en la distribución de los datos. Ciertamente, Julia no cuenta aún con muchas opciones para implementar estos métodos, pero es altamente probable que continúe el desarrollo de paquetes como *RobustStats*.

Respecto de los métodos de enseñanza y aprendizaje, por la misma naturaleza de la estadística como disciplina, se hace necesario articular otras opciones para el trabajo dentro y fuera del aula. En el Anexo 1, se describe un ejemplo de un caso de estudio, cuyo planteamiento fue generado por un grupo de alumnos universitarios que fueron guiados por su profesor. En el artículo se propone un método para implementar el Aprendizaje Basado en Desafíos (ABD) a través del cumplimiento de lineamientos que se incorporan de otros enfoques como el Aprendizaje Basado en Problemas (ABP), el Aprendizaje Basado en Proyectos (ABP) y el Aprendizaje Basado en la Investigación (ABI).

Es recomendable adherirse al comité de desarrollo del paquete de Julia, *RobustStats*, además de actualizar sus funciones para su uso en la versión 1.2. También, es necesario realizar un estudio de mayor alcance donde se someta a prueba el enfoque de ABD en una muestra grande ( $n > 390$ ) de alumnos universitarios.

## Bibliografía

- 
- [1] Akella, D. (2010) Learning together: Kolb's experiential theory and its application. *Journal of Management and Organization*, 16(1):100?112.



- [2] Bastide, P. and Solis-Lemus, C. and Kriebel, R. and Sparks, K. and Ané, C. (2018) Phylogenetic comparative methods on phylogenetic networks with reticulations. *Systematic Biology*.
- [3] Bezanson, J., Karpinski, S., Shah, V., and Edelman, A. (2012) Julia: A Fast Dynamic Language for Technical Computing. *arXiv preprint arXiv:1209.5145*.
- [4] De Haro, J. (2018) *Introducción a la programación con Julia*: Amazón México.
- [5] Ferrero, R. (2019) *Master en Estadística Aplicada con R Software: Tema 2*. [PDF] Descargado de <http://www.maximaformacion.es>
- [6] Flach, M., Gans, F., Brenning, A., Denzler, J., et al. (2017) Multivariate anomaly detection for Earth observations: a comparison of algorithms and feature extraction techniques. *Earth System Dynamics*, 8(3), 677-696.
- [7] Hampel, F. (1968) Contributions to the theory of robust estimation. *Phd, University of California, Berkeley, EUA*.
- [8] Hampel, F. (1971) A general qualitative definition of robustness of tests. *Annals of Mathematical Statistics*, 6(42), 1887-1896.
- [9] Hampel, F., Ronchetti, E., Rousseeuw, P. & Stahel, W. *Robust statistics: the approach based on influence functions*. John Wiley & Sons.
- [10] Huber, P. (1964) Robust Estimation of a Location Parameter. *Ann. Math. Statist.*, 35(1):73-101, 03 .
- [11] Huber, P. (1965) A Robust Version of the Probability Ratio Test. *Ann. Math. Statist.*, 36(6):1753-1758, 12.
- [12] Huber, P. (1972) The 1972 Wald Lecture Robust Statistics: A Review. *Ann. Math. Statist.*, 43(4):1041-1067, 08.
- [13] Huber, P. (1973) Robust Regression: Asymptotics, Conjectures and Monte Carlo. *Ann. Statist.*, 1(5):799-821.
- [14] Huber, P. (1977) *Statistical Decision Theory and Related Topics, chapter Robust covariances*. Academic Press.
- [15] Huber, P. (1981) *Robust statistics*. Wiley.
- [16] Huber, P. (2002) John W. Tukey's contributions to robust statistics. *Ann. Statist.*, 30(6):1640-1648, 12. Descargado de <https://doi.org/10.1214/aos/1043351251>
- [17] IASC. (2015) *Statutes of the International Association for Statistical Computing*. Descargado de <http://iasc-isi.org/statutes/>.
- [18] Kolb, D. (1984) *Experiential learning: experience as the source of learning and development*. Prentice Hall, New Jersey, EUA.
- [19] Nagar, S. (2017) *Beginning Julia Programming*. Apress, Berkeley, CA, New York, USA.
- [20] Pérez, A. (2005) *Métodos avanzados de estadística aplicada*. UNED.
- [21] Popuri, S. & Gobbert, M. (2017) *A Comparative Evaluation of Matlab, Octave, R, and Julia on Maya*. Technical report, Technical Report HPCF-2017-3, UMBC High Performance Computing Facility.
- [22] Sargent, T. & Stachurski, J. (2018) *QuantEcon. lectures - julia*. PDF. Lectures in Quantitative Economics. Descargado de <https://lectures.quantecon.org/jl/>.
- [23] Sawtzki, G. (2009) *Computational Statistics: An Introduction to R*. Chapman & Hall/CRC Press, Florida, USA.
- [24] Solano, J. (2011) *Introducción a la programación en Python*. Editorial Tecnológica del ITCR: Costa Rica.
- [25] Voulgaris, Z. (2018) *Julia for Data Science*. Technics Publications, NJ, USA.
- [26] Weibezahn, J., & Kendziorowski, M. (2019) Illustrating the Benefits of Openness: A Large-Scale Spatial Economic Dispatch Model Using the Julia Language. *Energies*, 12(6).

[27] Wickham, H & Grolemond, G. (2016) *R for Data Science: import, tidy, transform, visualize, and model data*. O'Reilly Media, Inc.

[28] Wilcox, R. (2016) *Introduction to Robust Estimation and Hypothesis Testing*. Academic Press, Londres.

## Anexo 1

**Planteamiento:** El sobrepeso y obesidad constituyen un problema de salud pública en naciones de los cinco continentes. La Organización Mundial de la Salud (OMS) estima que más del 30% de la población en América del Norte es obesa. En el caso del sobrepeso, las cifras son aproximadas. Existen diversas hipótesis bajo estudio para explicar el fenómeno, entre ellas sobresalen carga genética, alimentación inapropiada, estilo de vida sedentario, funcionamiento del sistema glandular (Tiroides), entre otras hipótesis más que se examinan de forma regional. Durante los últimos 20 años, el problema se ha extendido a todos los rangos de edad. Adultos mayores (60 años o más), adultos jóvenes, adolescentes e incluso niños forman parte del espectro que preocupa a la sociedad y las instancia gubernamentales. Por otra parte, la comunidad médica internacional y la (OMS) están valorando la pertinencia de seguir utilizando el Índice de Masa Corporal (IMC) como modelo para clasificar el estado de un ser humano. Así, otros modelos como los Índices de Cadera-Cintura (ICC), Cintura-Altura (ICA), Grasa Corporal (IGC) e Índice de Grasa Visceral (IGV) están sometiéndose a pruebas exhaustivas para articularse y diagnosticar de manera más precisa.

**Interrogantes:** ¿Cuáles son índices de mayor uso en medición corporal? ¿Qué proporción de estudiantes tienen sobrepeso de acuerdo con la escala de la OMS? ¿Qué proporción de estudiantes tienen obesidad grado 1, de acuerdo con la escala de la OMS? ¿Qué proporción de estudiantes tienen un IMC dentro de los parámetros que establece la OMS? ¿En qué medida la cantidad de estudiantes tienen un Índice Cadera-Cintura (ICC), Índice de Cintura-Altura (ICA), Índice de Grasa Corporal (IGC), e Índice de Grasa Visceral (IGV) dentro de los valores normales que establece la OMS? ¿En qué grado existe una correlación entre los IMC, ICC, ICA, IGC e IGV de los participantes?

**Objetivo:** El propósito es estimar el grado de sobrepeso y obesidad existente en un grupo de 210 sujetos (73 del sexo femenino y 137 del masculino), seleccionados al azar, en una institución universitaria. Los datos deben agruparse en 10 variables, se compararán los resultados a la luz de la aplicación de cuatro índices y es necesario redactar un informe reproducible \_ en términos computacionales\_ de sus resultados.

**Pertinencia:** Es necesario explorar en qué medida está presente el problema de sobrepeso y obesidad en poblaciones universitarias con edades entre los 18 y 25 años. También, es indispensable examinar una situación problemática desde otras perspectivas metodológicas y comparar la eficacia de los diferentes modelos que se aplicarán.

**Hipótesis:** En función de los estándares establecidos por la OMS y la revisión de la literatura publicada a partir del año 2015, se plantearon las conjeturas siguientes en relación con los sujetos participantes:

$H_1$  Al menos el 5% tienen obesidad mórbida.

$H_2$  Menos del 10% tienen un IMC inferior al normal.

$H_3$  Entre el 50 y 60% tienen un IMC dentro de los límites normales.

$H_4$  Más del 60% tienen un ICC e ICA dentro de los límites normales.

$H_5$  No existen diferencias significativas entre los diferentes índices manuales para medición corporal.

**Marco de referencia:** La realización del estudio se fundamentará en los lineamientos teóricos y metodológicos establecidos por la OMS y la International Obesity Task Force (IOFT). Las categorías de análisis tendrán que ser acordes a dichos lineamientos.

**Método propuesto:** La naturaleza del caso es el punto de partida para seleccionar los instrumentos y el procedimiento por seguir. Las interrogantes planteadas y el objetivo del estudio exigen utilizar un método cuantitativo.

Se propone un estudio descriptivo, transversal y comparativo. A cada sujeto se le medirá su estatura, el diámetro del cuello, torso, cintura, cadera, tobillo, bíceps y pantorrilla. Se obtendrá su peso corporal y edad.

Como se indicó líneas atrás, participarán 210 sujetos seleccionados al azar a través de un muestreo probabilístico por estratos (sexo). La cantidad de participantes del sexo femenino y masculino están acorde con las proporciones de la población. En esta etapa del estudio no se manipulará ninguna variable de forma deliberada. Únicamente interesa observar, describir e interpretar medidas en un momento único.

## Anexo 2.1

---

```

using DataFrames, CSV, StatsBase, Statistics, Distributions, HypothesisTests, Plots, StatsPlots, Gadfly, RobustStats

#Datasets = CSV.read("Datasets.csv", header = true)
EdadFem = [19,19,19,19,21,19,19,19,20,21,19,18,20,17,21,22,19,24,18,21,26,20,20,20,24,21,21,22,22,19]
EstaturaFem = [1.62,1.62,1.7,1.63,1.54,1.69,1.74,1.54,1.59,1.74,1.7,1.81,1.67,1.69,1.63,1.54,1.59,1.56,1.67,1.67,1.66,1.57,1.67,1.52,1.64,1.6,1.63,1.58,1.58,1.64]
EdadMas = [20,20,23,20,19,19,19,23,21,19,19,20,20,19,20,22,18,19,21,21,19,21,22,19,19,18,19,18,20,20]
EstaturaMas = [1.73,1.8,1.72,1.66,1.8,1.79,1.72,1.71,1.83,1.76,1.71,1.77,1.65,1.75,1.74,1.7,1.71,1.74,1.65,1.8,1.75,1.8,1.73,1.77,1.93,1.73,1.79,1.76,1.73,1.7]

# Análisis Exploratorio de Datos

using FreqTables
tabla1 = freqtable(EdadFem,EdadMas)
tabla2 = freqtable(EstaturaFem, EstaturaMas)

## Representaciones gráficas de algunas variables

gr()
H1 = histogram(EdadFem, nbins = :sturges,leg = false,normed = true,tittle = "Sexo femenino", xlab = "Edades",ylab = "Frecuencia", color = "pink")
H2 = histogram(EdadMas, nbins = :sturges,leg = false,normed = true,tittle = "Sexo Masculino",xlab = "Edades",ylab = "Frecuencia")
plot(H1, H2, layout = (1,2), leg = false)

```

Figura 1.5: Código de ejemplo

```
# Estatura sexo masculino
AjusteMas = fit(Normal, EstaturaMas)
variableMas = EstaturaMas
NMas = Gadfly.plot(x = AjusteMas, y = variableMas, Stat.qq, Geom.point
)

# Estatura sexo femenino
AjusteFem = fit(Normal, EstaturaFem)
variableFem = EstaturaFem
NFem = Gadfly.plot(x = AjusteFem, y = variableFem, Stat.qq, Geom.point
)

REdadFem = summarystats(EdadFem)
REdadFem = summarystats(EdadFem)
REdadFem = summarystats(EdadFem)
REstaturaMas = summarystats(EstaturaMas)

## Ejemplos de pruebas estadísticas

using HypothesisTests
OneSampleADTest(variableMas, Normal())

using RCall
R"shapiro.test($variableMas)"

## Métodos robustos

tmean(EstaturaFem, tr=0.2)
winmean(EstaturaFem, tr=0.2)
tauloc(EstaturaFem)
onestep(EstaturaFem)
mom(EstaturaFem)
bisquareWM(EstaturaFem)
huberWM(EstaturaFem)
trimean(EstaturaFem)

winvar(EstaturaFem, tr=0.2) -
wincov(EstaturaFem, y, tr=0.2)
tauvar(EstaturaFem)
igrn(EstaturaFem)

outbox(EstaturaFem)
```

Figura 1.6: Código de ejemplo (continuación).