

IMPLEMENTACIÓN DE UN SCRIPT PARA ESTIMACIÓN VOLUMÉTRICA A TRAVÉS
DE MODELOS DE AHUSAMIENTO EN PYTHON Y R PROJECT.

Ortiz Barajas Jofrain Alonso, Parra Prieto Bryan Daniel

Trabajo de Grado para Optar al Título de Ingeniero Forestal

Director

Rony Alexander Ortiz Aponte.

Ingeniero Civil

Universidad Industrial de Santander

Instituto de Proyección Regional y Educación a Distancia - IPRED

Programa de Ingeniería Forestal

Bucaramanga

2020

Dedicatoria

En primer lugar darle las gracias a Dios todo poderoso por la salud y la vida que me ha dado cada día para desarrollar este proyecto y culminarlo con gran satisfacción, también darle gracias a mis padres Guillermo Ortiz y Gladys Barajas por el apoyo incondicional al empezar y al terminar mis estudios universitarios en donde les dedico este triunfo a ellos por gran esfuerzo que han dado por mí, agradezco también a mis amigos, compañeros y familiares por los ánimos y acompañamiento que me han dado en todo tiempo, también doy gracias al mejor amigo y compañero de tesis que es Daniel Parra por la paciencia y el esfuerzo que tuvimos en el desarrollo del proyecto.

Alonso Ortiz.

- Principalmente a Dios, sin su benevolencia y fortaleza no hubiese llegado hasta este punto.
- A mi padre y abuela, Miguel Parra y Rubiela Estupiñan, por su amor, esfuerzo continuo, amparo y ética inculcada desde niño.
- A mis hermanas, hermanos, madre y demás familiares por todo el apoyo y moral que siempre me han brindado.
- A los maestros, quienes sentaron los cimientos para mi formación profesional.
- A todos mis amigos, docentes y demás personas que de alguna u otra manera han aportado en mi proceso de formación como persona.
- A mi compañero Alonso Ortiz por permitirme trabajar de manera conjunta en la ejecución de este proyecto.
- A mi gran amigo Arley Alvarado quien se ha convertido como en un hermano.

Daniel Parra

Agradecimientos

A la Universidad Industrial de Santander por acogernos y convertirse en nuestra alma máter.

Al Dr. Jorge Andrés Rodríguez Toro por la dirección, acompañamiento y enseñanza durante la etapa académica y ejecución del proyecto de grado.

Al Ingeniero Rony Alexander Ortiz Aponte por aceptar la dirección de este proyecto.

Absolutamente a todas las personas que han hecho parte de nuestra vida ya sea de manera directa e indirecta.

“El objetivo de la educación es la virtud y el deseo de convertirse en un buen ciudadano”

Platón

Tabla de Contenido

Introducción	12
1. Objetivos	14
2. Marco Referencial	15
2.1. Marco Teórico	15
2.2. Marco Conceptual	18
2.3. Marco legal	25
3. Metodología	26
3.1. Adquisición de la información	26
3.2. Instalación de los lenguajes de programación	27
3.2.1. Instalación de Python	27
3.2.2. Instalación de R-Project	31
3.3. Tabulación y análisis de datos	37
3.4. Ajuste de los modelos de ahusamiento	42
3.5. Obtención del volumen	44
3.5.1. Estimación del volumen a través de integración numérica	44
3.5.2. Estimación del volumen por método analítico	45
3.6. Bondad de ajuste	48
3.6.1. Supuestos básicos de regresión	49
3.7. Evaluación de la precisión del ajuste del modelo	49
3.8. Interfaz gráfica para obtener el factor de forma de los individuos	50

3.9. Código QR	51
3.10. Formato de etiqueta del código QR	52
3.11. Resumen del script generado en R-Project y Python	52
3.11.1. Fase estadística y matemática computacional	53
3.11.2. Fase en R-Project	53
3.11.3. Fase en Python	54
3.12. Valor agregado	54
4. Resultados	55
5. Recomendaciones	81
6. Conclusiones	82
Referencias Bibliográficas	84
Apéndices	89

Lista de Figuras

Figura 1. Página oficial de Python.....	27
Figura 2. Descarga del instalador de Python	28
Figura 3. Editar las variables de entorno del sistema.....	29
Figura 4. Añadiendo la ubicación de Python al sistema	30
Figura 5. El Sistema reconociendo a Python.	31
Figura 6. Página oficial de R.....	32
Figura 7. Espejo o sitio de distribución para R-Project	32
Figura 8. Descarga del instalador de R-Project.	33
Figura 9. Opción de descarga de las librerías en R-Project.	34
Figura 10. Sitio de distribución de R-Project.....	35
Figura 11. Seleccionando las librerías con las que trabaja R-Project.....	36
Figura 12. Instalación de paquetes en R-Project a través de código.....	37
Figura 13. Ejemplo de una forma genérica fustal de un árbol. Adaptado de (Omonte & Valenzuela, 2011).	44
Figura 14. Perfil de sólidos geométricos básicos. Adaptado de (Cancino, 2006).	46
Figura 15. Sólidos de revolución: cilindro, paraboloides, cono y neiloide. Adaptado de (Riaño Melo & others, 2017).	47
Figura 16. Interfaz gráfica obtenida en Python.....	51
Figura 17. Formato de etiqueta código QR.....	52

Figura 18. Configuración del equipo.	89
Figura 19. Configuración del equipo.	90
Figura 20. Notebook Jupyter R-Project.	91
Figura 21. Corriendo las líneas.	91
Figura 22. Subida de la base de datos.	92
Figura 23. Notebook «Jupyter Python.	93
Figura 24. Instalación de los módulos en la consola del cmd para generar el código QR..	94
Figura 25. Entorno Jupyter.	96
Figura 26. Herramienta Jupyter.	97
Figura 27. Entorno del primer trabajo.....	97
Figura 28. Códigos en Jupyter.	98

Lista de Tablas

Tabla 1. Comparación de aspectos generales entre los programas estadísticos SPSS, SAS y R-Project.	16
Tabla 2. Tipos de modelos de regresión	22
Tabla 3. Variables básicas del inventario forestal tabuladas en una hoja de cálculo.....	38
Tabla 4. Duplicidad de datos para efectuar la convergencia de los parámetros de los modelos ..	40
Tabla 5. 70 % de los datos para realizar el ajuste	41
Tabla 6. Declaración de las variables dependientes e independientes de los modelos de ahusamiento para su respectivo ajuste	42
Tabla 7. Modelos de ahusamiento contenidos en del desarrollo del script.....	43
Tabla 8. Ecuaciones de cubicamiento a través de métodos analíticos tradicionales.	47
Tabla 9. Información almacenada en el código QR.....	51
Tabla 10. Requisitos mínimos.....	95

Lista de Apéndices

Apéndice A. Manual del Programador 89

Apéndice B. Manual del usuario..... 95

Resumen

Título: implementación de un script para estimación volumétrica a través de modelos de ahusamiento en python y r-project *

Autor: Jofrain Alonso Ortiz Barajas, Bryan Daniel Parra Prieto **

Palabras Clave: Modelo no lineal, Programación, Algoritmo, Bondad de ajuste, Residuales.

Descripción:

Muchas veces efectuar operaciones de índole estadístico y matemático puede resultar en algo complejo sobre todo cuando se trabaja con grandes bases de datos o registros, para realizar esta labor en conjunto con el avance de las tecnologías informáticas la habilidad de programar se vuelve una herramienta trascendental para el análisis de datos respectivamente, a continuación se presenta el primer script generado en lenguajes de programación bajo distribución gratuita (Python y R-Project) en la Universidad Industrial de Santander sede Málaga, que ajusta nueve modelos de ahusamiento para especies de rápido crecimiento (pinos y eucaliptos) a través de la obtención de los parámetros de los modelos empleando el algoritmo de Gauss Newton y Levenbert Marquardt, bondad de ajuste, comportamiento gráfico del perfil longitudinal fustal, aptitud del modelo evaluado mediante el coeficiente de determinación (R^2), gráfico de los supuestos de regresión, precisión de los modelos evaluada mediante la raíz del error cuadrático medio (RMSE), el modelo de ahusamiento que presente mejor ajuste se integra de manera numérica y dicho volumen obtenido es contrastado con el valor observado obtenido por métodos analíticos, posteriormente dicha información volumétrica es almacenada en un código QR en conjunto con otras características (nombre científico, nombre común, familia, especie, coordenadas, estado fitosanitario, manejo, usos, volumen estimado, modelo empleado, factor de forma), con el objetivo de caracterizar el estado del rodal y así visualizar la planeación y gestión del sector para su respectivo aprovechamiento forestal integral.

* Trabajo de grado

** Instituto de Proyección Regional y Educación a Distancia - IPRED. Programa de Ingeniería Forestal. Director: Rony Alexander Ortiz Aponte, Ingeniero Civil.

Abstract

Title: implementation of a script for volumetric estimation through taper models in python and r-project *

Author: Jofrain Alonso Ortiz Barajas, Bryan Daniel Parra Prieto **

Keywords: Nonlinear model, Programming, Algorithm, Goodness adjustment, Residual.

Description:

Many times to perform statistics and mathematic operations may result in something complex especially when you work with large database or registers, to do this work in conjunction with the advancement of computer technologies the ability to program becomes a momentous tool for data analysis respectively, below is the first script generated in programming languages under free distribution (Python y R-Project) at Universidad Industrial de Santander Headquarters Malaga, which adjust 9 taper models for fast-growing species (pines and eucalyptus trees) through obtaining the parameters of the models using the Algorithm of Gauss Newton and Levenbert Marquardt, goodness of adjustment, graphic behavior of the right longitudinal fustal, attitude of the model assessed by coefficient of determination (R^2), graphic of the regression assumption, accuracy of the models assessed by the the root mean squared error (RMSE), the taper model that presents the best whole adjustment numerically and that volume obtained is contrasted with the observed value obtained by analytical methods, subsequently such volumetric information is stored in a QR code in conjunction with other features (scientific name, common name, family, specie, coordinates, phytosanitary status, management, model, uses, stimated volume, form factor), in order to characterize the state of the stand and thus visualize planning and procedure of the sector for their respective integral forestry use.

* Bachelor Thesis

** Instituto de Proyección Regional y Educación a Distancia - IPRED. Programa de Ingeniería Forestal. Director: Rony Alexander Ortiz Aponte, Ingeniero Civil.

Introducción

A nivel educativo especialmente en la educación superior, la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO, 1998), ha estimado a las Tecnologías de la Comunicación e Información (TIC) como elementos fundamentales en la realización de procesos de innovación curricular dentro de la educación superior que permita facilitar el acceso de una mayor población a esta etapa (Mayo & Álvarez, 2011). Desde que se generó el desarrollo de las TIC hoy en día éstas han venido evolucionando a pasos gigantescos, situaciones no ajenas al campo de la ingeniería y el mundo empresarial que debe estar a la vanguardia si desea adaptarse al ritmo de la sociedad contemporánea y por ende de la competencia (Betancur Calle & Pérez García, 2013).

Muchas veces realizar operaciones sobretodo de índole matemático, estadístico o de simple representación visual puede ser algo extenuante, si bien es cierto que la programación lineal es el método de optimización más distribuido y utilizado en diversos sectores de la ingeniería, la economía, el manejo de recursos naturales, entre otros, desde una perspectiva global en el sector forestal no es tan frecuente su uso en nuestro país; el crecimiento exponencial de la informática y el fácil acceso al trabajo diario con ordenadores, hacen de esta técnica un instrumento eficaz para la gestión de las empresas y los productores del sector (Coronel de Renolfi, 2004).

Actualmente es válido afirmar que la relación entre métodos estadístico-matemáticos y los grandes avances de la informática son en conjunto los elementos más importantes de la mensura forestal, este último concepto se interpreta como la ciencia encargada de la medición del bosque y sus productos (Prodan, 1997). En las ciencias forestales una de las actividades más importantes es el aprovechamiento maderable del bosque o de plantaciones comerciales (Návar-Cháidez &

Domínguez-Calleros, 2013) en esta se debe calcular de manera precisa la cantidad del producto ya que los grandes volúmenes de las masas arbóreas pueden generar limitantes de manejo si se desconoce su cantidad, y si no se emplean modelos matemáticos de alta precisión que incidan a la toma de decisiones en la gestión forestal (Clutter, Fortson, Pienaar, Brister, & Bailey, 1983).

En la carrera de pregrado de Ingeniería Forestal de la Universidad Industrial de Santander Sede Málaga actualmente el tema de la programación lineal ha sido una limitante debido al poco vínculo existente entre el uso de tecnologías informáticas y la aplicabilidad práctica en las asignaturas. En el siguiente trabajo se describe el desarrollo de un script empleando los lenguajes de programación Python y R-Project, que permite cubicar individuos sin efectuar muestreos destructivos a través de información contenida en hojas de cálculo referentes a inventarios forestales, tomando variables de alta importancia como diámetro a la altura del pecho (DAP) y altura total del individuo a través del ajuste de nueve modelos de ahusamiento que en un principio alimentan el código pero que a futuro se espera que sea retroalimentado tanto por modelos lineales, no lineales, segmentados y trigonométricos, de tal forma que se genere un algoritmo que contenga la mayor cantidad de modelos posibles que ajusten tanto a especies exóticas como nativas.

1. Objetivos

1.1. Objetivo General

Desarrollar e implementar un script en los lenguajes de programación Python y R-Project que permita estimar el volumen de individuos forestales a través de modelos de ahusamiento.

1.2. Objetivos Específicos

Ajustar y comparar diversos modelos de ahusamiento en el lenguaje de programación R-Project que de acuerdo a la convergencia de la totalidad de los parámetros, bondad de ajuste, y comportamiento gráfico fustal determine la ecuación más idónea a seleccionar en referencia a la base de datos forestal.

Validar los datos obtenidos del volumen a través del factor de forma en una interfaz gráfica y almacenar dicha información en un módulo QR mediante el lenguaje de programación Python.

2. Marco Referencial

2.1. Marco Teórico

¿Qué es R-Project?

R-Project es un lenguaje de programación interpretado, de distribución gratuita, bajo Licencia GNU, cuya fortaleza está ambientada para el cómputo estadístico y gráfico. La terminología ambientada aspira a caracterizarlo como un software totalmente estructurado y coherente, en cambio de una serie de acumulaciones progresivas de herramientas muy definidas y poco flexibles, como suele suceder con otros software de análisis de datos (Santana & Mateos, 2014).

Fue fundado en el año 1992 en Nueva Zelanda por Ross Ihaka y Robert Gentleman cuya primera intención era crear un lenguaje comprensible, que pudiera ser utilizado en el preámbulo de los cursos de estadística en la Universidad de Auckland - Nueva Zelanda (Santana & Mateos, 2014).

R Project es un lenguaje de programación de estadística descriptiva en el cual se desarrollan diferentes operaciones matemáticas dando resultados en poco tiempo y facilitando el rendimiento de tiempo en su desarrollo, este lenguaje de programación es constantemente actualizado por una comunidad de profesionales.

El nombre “R” sale de la idea original de las iniciales tanto de Ross como de Robert (sus fundadores) y hasta el día de hoy es el nombre con el que se le conoce a escala mundial y científica. A continuación, en la tabla 1 se resume la versatilidad característica ofrecida por el lenguaje de programación “R” comparado con los programas estadísticos SAS Y SPSS, estos últimos muy citados en el ámbito académico y científico (Salas, 2008).

Tabla 1.

Comparación de aspectos generales entre los programas estadísticos SPSS, SAS y R-Project.

Aspecto	SPSS	SAS	R-Project
Amigabilidad con el usuario	Excelente	Baja-Regular	Baja-regular
Manipulación de datos	Baja	Buena	Buena
Calidad de gráficos	Regular	Buena-Excelente	Excelente
Control de procesos	Baja	Excelente	Excelente
Costo	U\$S 1500	U\$S 7200	Gratis
Código fuente disponible	No	No	Si
Variedad análisis estadísticos	Buena	Buena-Excelente	Excelente
Documentación	Excelente	Buena	Bajo
Soporte técnico	Bueno	Bueno	Bajo
Sistema operativo	Windows	Windows, Macintosh, Linux	Windows, Macintosh, Linux

¿Qué es Python?

Python es un lenguaje de programación que fue creado a principio de los años 90 por Guido van Rossum, el nombre de este lenguaje de programación se toma en base a un grupo de cómicos ingleses denominado “Monty Python”; Python posee una sintaxis amplia y sencilla que favorece al programador con códigos legibles (Duque, 2011). Este entorno de programación se revela en que con tan solo unas pocas líneas de código se pueden ejecutar actividades (de cómputo) complejas que en otro lenguaje de programación requerirían de más líneas de código (o mayor número de instrucciones) (Rojas, Fernández, & Ruiz, 2016).

Python es un lenguaje programación que permite con gran facilidad y velocidad desarrollar o automatizar algún script (programa) para solucionar una tarea en específico, este lenguaje de programación posee una sintaxis simple y sencilla para el usuario que se inmerge en el deseo de aprender a programar, dicho lenguaje está disponible para sistemas operativos como Windows, Mac OS X y Unix (Rossum, Drake Jr, & Fred, 2009).

Python es un lenguaje poderoso por lo cual se puede extender a nuevas funcionalidades en el desarrollo de una tarea y manejar tipos de datos implementados en C y C++; en donde se pueden elaborar script (programa) o líneas de código con el objetivo de cumplir alguna tarea en el sistema operativo que se esté trabajando ya sea Windows, Mac OS X y Unix más rápidamente que C, C++ o Java (Rossum et al., 2009).

LyX

Es un programa gratuito que fue diseñado para crear documentos profesionales, este programa se presta para crear archivos como informes, cartas, libros, entre otros documentos. Este programa se encuentra disponible para sistemas operativos como Windows, Mac Os y Linux, su descarga se puede realizar desde el siguiente link: <https://www.lyx.org/Download>

Nota: Una vez descargado el programa e instalado, es necesario instalar los paquetes que se necesitan para el funcionamiento de LYX, estas se pueden descargar desde la consola de MiKTeX.

Conda, es un paquete de Python el cual es utilizado por la versatilidad y la funcionalidad para trabajos más avanzados en donde este paquete trae incluido una serie de Entornos de Desarrollo Integrados (IDE) muy útiles. Se puede descargar este software desde el siguiente link: <https://www.anaconda.com/distr>

Funciones de ahusamiento

Las funciones o modelos de ahusamiento se pueden denominar como aquellas ecuaciones que describen la forma geométrica del árbol (cono, parábola, entre otros), entendido entre la rata del decrecimiento del diámetro a medida que aumenta la altura fustal (Prodan, 1997).

Los modelos de ahusamiento resultan una herramienta estadística trascendental de utilidad corriente en la estimación de volúmenes tanto de árboles como de trozas, permitiendo tres características básicas (Prodan, 1997), descritas a continuación:

- Diámetro en cualquier punto del fuste.
- Altura del fuste a la que se encuentra un diámetro límite específico.
- Volumen entre dos puntos cualesquiera del fuste.

A partir del siglo XIX se han venido analizado diversos métodos que permitan expresar la forma y ahusamiento de los individuos arbóreos relacionando las variables diámetro-altura. Con el avance y desarrollo de las herramientas computacionales se traspasó de métodos gráficos a métodos analíticos, desarrollando con el pasar de los años múltiples modelos (Prodan, 1997).

Los modelos de ahusamiento son diversos métodos matemáticos que se utilizan para la estimación de volumen de un árbol por método destructivo, calculando el volumen de la madera con corteza y sin corteza por medio de rodela a diferentes alturas y diámetros del fuste, por eso son una herramienta importante para el cálculo puntual de las estimaciones de volumen de madera, también es importante mencionar que estos modelos matemáticos sirven para el desarrollo de simulaciones que optimizan la madera para el uso comercial (Rodríguez Toro et al., 2016).

Según Rodríguez Toro et al. (2016) sugiere que los modelos de ahusamiento deben tener los siguientes atributos: “i) ser diferenciable en toda la longitud del fuste, ii) no generar oscilaciones alrededor de la línea de tendencia central, iii) que estime de manera adecuada no solo el diámetro de la sección comercial del fuste, sino también el diámetro del tocón para predecir el diámetro normal y el volumen del árbol para la reconstrucción de las variables de estado del rodal en un escenario postcosecha y el diámetro del fuste por encima del límite de utilización comercial para aserrío y, iv) la capacidad del modelo de ahusamiento para predecir la altura del fuste, a la que se halla un diámetro límite de utilización para fines comerciales”.

2.2. Marco Conceptual

Software libre: Se define como software libre por la facilidad de adquirirlo, porque es gratuito y su ventaja es que hay transparencia en las evaluaciones que se le hacen al desarrollo de los datos (Carita, 2018).

Ciencia de datos: De acuerdo con Rogel-Salazar (2018) define la Ciencia de datos como «la extracción del conocimiento y la comprensión de diversas fuentes de datos, y las habilidades necesarias para lograr este rango, desde la programación hasta el diseño, y desde las matemáticas hasta la narración de historias».

Algoritmo: Son métodos y notaciones matemáticas empleadas para hallar una solución a un problema (Pacheco, 2019). En el lenguaje de la programación y la informática el algoritmo se define como un conjunto de instrucciones secuenciales que llevan a cabo ciertos procesos para definir un resultado, tomar ciertas decisiones o determinar algunas de las necesidades planteadas (Raffino, 2019).

Forma de presentar datos: Es un método en donde se analizan todos los datos procesados por medio de una presentación gráfica, la cual permite comprender y analizar la información que aportan, también comprender el comportamiento de los mismos (Correa, 2002).

Según Correa (2002) estas observaciones gráficamente pueden aportar los siguientes beneficios:

1. Simplificar lo tedioso y lo complejo.
2. Brindar ayuda visual.
3. Un gráfico es más accesible que una tabla.
4. Es una ayuda para el cerebro, ya que permite entender y memorizar mejor.

Variable cuantitativa: Son variables medibles, cuyos datos se representan por grados o intensidades, las cuales tiene un objetivo que es el carácter numérico; como por ejemplo los ingresos, pesos, alturas, entre otros (Cauas, 2015).

Criterio de Información Akaike (AIC): Es una combinación en la información teórica, la entropía y la verosimilitud de los datos (López et al., 2016), en donde se define la siguiente ecuación:

$$AIC = 2\log Lik + 2K$$

LogLik = logaritmo de probabilidad

Según López et al. (2016) describe que «Este criterio tiene en cuenta los cambios en la bondad de ajuste y las diferencias en el número de parámetros entre dos modelos. Los mejores modelos fueron aquellos que presentaron el menor valor de AIC».

Este criterio es útil para la elección de modelos que presenta variaciones en el número de parámetros, en el ajuste de modelos cuya información sea relativamente igual, este criterio permitió la selección del modelo más simple (Cancino, 2006).

Criterio de información Bayesiano (BIC): De acuerdo con López et al. (2016) El BIC es calculado para los diferentes modelos como una función de la bondad de ajuste del log Lik (logaritmo de probabilidad), el número de parámetros ajustados y el número total de datos. El modelo con el más bajo valor de BIC es considerado el mejor en explicar los datos con el mínimo número de parámetros», en donde se define la siguiente ecuación:

$$BIC = -2\log Lik + \log(N) * K(6)$$

Regresión lineal simple: El objetivo de un modelo de regresión lineal simple es el de explicar la relación que existe entre una variable dependiente o de respuesta y una única variable dependiente o explicativa (Limeres, 2012).

Supuestos básicos de regresión: La metodología más usada en el análisis de regresión es la del método de mínimos cuadrados, este método en términos computacionales es realmente simple y,

cuando satisface ciertos supuestos, expone varias propiedades estadísticas que lo hacen eficaz (Gujarati, 1981).

Por tanto:

1. Debe existir una aleatoriedad de los residuales.
2. Los residuos presentan una distribución normal.
3. Homocedasticidad de varianza.
4. Los residuos tienen varianza σ^2 constante.

El modelo de regresión es bastante robusto, por lo que particularmente no es necesario que se cumplan en su totalidad las condiciones anteriores sobre todo la 3 y la 4 (Ramirez-Alan, 2017)

Análisis de correlación: Según Aparicio et al. (2004) un análisis de correlación permite cuantificar el grado de asociación lineal entre variables, y en particular, entre las variables explicativas continuas disponibles y la respuesta de interés.

Coefficiente de determinación: Es un índice utilizado para analizar el tipo de bondad que tiene un modelo de regresión lineal al ser ajustado a una situación en específico (Jáuregui & Oliden, 2004).

Modelos no lineales: Pueden describirse como una ecuación que describe la relación no lineal entre una variable de respuesta y una o más variables predictoras (Minitab, 2020), estimando nuevas observaciones, dependiendo de la forma de esta función se encuentran la siguientes descritas en la tabla 2 (Sánchez Alberca, 2014):

Tabla 2.

Tipos de modelos de regresión

Modelo	Ecuación genérica
Lineal	$y = a+bx$
Parabólico	$y = a+bx+cx^2$
Polinómico de grado n	$y = a_0+a_1x+\dots+a_nx^n$
Potencial	$y = ax^b$
Exponencial	$y = e^{a+bx}$
Logarítmico	$y = a+b\text{Log}x$
Inverso	$y = a+b/x$
Curva S	$y = e^{a+b/x}$

Algoritmo de Gauss-Newton: De acuerdo con Cornejo Zuniga and Rebolledo Vega (2016) “es un procedimiento que a través de ciertas propiedades de convergencia permite aproximar la matriz Hessiana en cada iteración presente mediante fórmulas de recurrencia en el modelo, demostrando ser eficiente en optimización lineal”.

$$x_{+} = x_c - (J(x_c)^T J(x_c)^{-1} J(x_c)^T R(x_c))$$

Algoritmo de Levenberg-Marquardt: Según Cornejo Zuniga & Rebolledo Vega (2016) establece que: “También conocido como el método de mínimos cuadrados amortiguados ha sido una técnica numérica estándar para casos de mínimos cuadrados no lineales frecuentemente aplicada en varias disciplinas para el ajuste de parámetros, siendo Levenberg – Marquardt una buena alternativa para evitar los problemas del método Gauss – Newton cuando la matriz del modelo no es singular”.

$$x_{+} = x_c - (J(x_c)^T J(x_c) + \mu_c I)^{-1} J(x_c)^T R(x_c)$$

Evaluación de la precisión del ajuste del modelo: De acuerdo con González (2018) “indica el ajuste absoluto a los datos, en cuán cerca están los puntos de datos observados de los valores predichos del modelo. La raíz del error cuadrático medio (RMSE) es una medida absoluta de

ajuste. Los valores más bajos de RMSE indican un mejor ajuste. RMSE es una buena medida de la precisión con que el modelo predice la respuesta, y es el criterio más importante para ajustar si el propósito principal del modelo es la predicción.”

$$\text{RMSE} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{(n-p)}}$$

Donde:

y_i = valores observados de la variable dependiente (estimación del volumen por método analítico)

\hat{y}_i = valores del volumen predicho obtenido a través de los modelos de ahusamiento

n = número de datos ajustados en el modelo

p = número de parámetros en el modelo

Operaciones numéricas: Son operaciones matemáticas que se desarrollan, en este caso a través de lenguajes de programación haciendo la misma función que una calculadora convencional como sumar, restar, multiplicar y dividir, entre otras de mayor complejidad (Rossum G. v., 2017).

Flotantes y enteros: Los flotantes son la representación de números que no son enteros como por ejemplo 0.5 o -7.3256 que son valores decimales, en Python se declaran como tipo float y los enteros son la representación de números no decimales que pueden ser positivos o negativos y se declaran como int (Rossum G. v., 2017).

String: Es una cadena de texto que se utiliza para ingresar una información o argumento utilizando dobles comillas, comillas sencillas o seguido de un numeral (Rossum G. v., 2017).

Operadores lógicos: Operadores que se utilizan en programación, son los booleanos y los de comparación; los booleanos es cuando se define en la sintaxis si las afirmaciones son verdaderas (True) o falsas (False) y los de comparación de variables (SoloLearn, 2019).

Estructuras de control: Es una forma secuencial en la que se establece una serie de condiciones utilizando las sentencias if que del inglés significa SI y else que significa que en el caso de no cumplirse la condición se ejecute la siguiente condición (SoloLearn, 2019).

Código QR: Es un tipo de código en el cual se almacena información codificada por medio de barras bidimensional (Laporte, Reulier, Ternoir, Chapuis, & Kassel, 2011), siendo la evolución del código de barras tradicional.

Script: Se refiere al código fuente escrito en algún tipo de lenguaje de programación interpretado.

Altura total: Es la distancia longitudinal que tiene un árbol desde la parte inferior (tocón) hasta la parte superior (ápice de la copa)

Apeo: Es un método operativo que consiste en cortar un árbol desde la parte inferior (tocón).

Aprovechamiento: Conjunto de todas las operaciones, incluidas la planificación previa y la evaluación posterior, relacionadas con el apeo de los árboles y el desembosque de sus troncos u otras partes aprovechables para su posterior transformación en productos industriales. Se denomina también aprovechamiento de madera.

Corteza: Es el conjunto de tejidos primarios y capas de fibra vegetal que envuelve la parte exterior de algunas plantas y árboles.

Cubicación: Determinación de la cantidad de troncos obtenidos, por lo común, midiendo el volumen (en maderos de pequeño tamaño, también por peso; esto último es normal en el caso de la madera para pasta; el pesaje se realiza en ese caso en el centro de elaboración).

DAP: Medida del diámetro tomada a la altura del pecho a 1.30 m.

Dasometría: Ciencia que trata de la medición de los bosques y de los árboles en donde se mide el volumen de todas las partes del árbol, la edad, las existencias de maderas en rodales y el incremento de árboles individuales y de rodales completos, así como la magnitud y volumen de sus productos.

Dendrometría: Significa medición de árboles e incluye la verificación y análisis de datos tales como la altura, diámetro, área basal, volumen, entre otros.

Fuste: Tronco de un árbol que alcanza un grosor notable de manera que de él pueden obtenerse trozas, rollos para chapa o postes de gran tamaño.

Individuo: Cada uno de los elementos de la población, en este caso hace referencia a cada árbol del estudio.

Población: Conjunto de todos los elementos que verifican una característica que será objeto de estudio.

Rodal: Agrupación de árboles u otras plantas que por alguna circunstancia particular se distingue del ecosistema que le rodea. Es suficientemente uniforme en la especie, edad, calidad o estado para poder distinguirse del arbolado que lo rodea.

Silvicultura: Es la ciencia que trata de la conservación y manejo de los bosques, con la finalidad de obtener de ellos el mayor provecho posible. Llamada también Dasonomía o Ecología Forestal.

Tiene que ver con el corte de árboles, con el objeto de obtener una regeneración natural o artificial de los bosques por medio de podas, producción de plantones en el vivero, plantaciones, entre otros.

Tocón: Base del árbol que queda en el suelo después del apeo.

Volumen en pie: Cantidad de madera en metros cúbicos cuando el individuo o masa forestal está en pie.

Volumen: Cantidad de espacio ocupado por un cuerpo.

2.3. Marco legal

- Ley 1273 del 2009: el cual establece “por medio de la cual se modifica el Código Penal, se crea un nuevo bien jurídico tutelado - denominado "de la protección de la información y de los datos" y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones”.
- Ley 23 de 1982 por medio del cual se expide sobre los derechos de autor establecido por el Congreso de Colombia el 28 de enero de 1982 donde se decreta desde las disposiciones generales sobre los derechos de autor hasta las disposiciones jurídicas en la protección, control y sanción de los derechos de autor.
- Decreto 1791 de 1996 por medio del cual se establece el régimen de aprovechamiento forestal.
- Decreto 1078 de 2015 por el cual se expide el decreto único reglamentario del sector de las tecnologías de la información y las comunicaciones.
- Ley 1680 de 2013 por la cual se garantiza a las personas ciegas y con baja visión, el acceso a la información, a las comunicaciones, al conocimiento y a las tecnologías de la información y de las comunicaciones.

3. Metodología

3.1. Adquisición de la información

Dado algunos aspectos logísticos y sobretodo económicos no se realizó un muestreo destructivo de individuos que es lo recomendado por la literatura, por el contrario las líneas de código programadas optan por el suministro de una base de datos realizada por los compañeros de la

asignatura “Mediciones Forestales” de VI nivel de la Universidad Industrial de Santander sede Málaga, reduciendo tanto costos como el tiempo empleado para realizar dichas labores, cabe aclarar que la información de un inventario forestal debe tener variables básicas como altura del individuo, diámetro a la altura del pecho (DAP), entre otros, lo cual no limita el alcance del script en lo absoluto.

3.2. Instalación de los lenguajes de programación.

A continuación se describe el proceso de instalación de los lenguajes de programación más usados actualmente en el tema de ciencia de datos (Python y R-Project) (INESEM, 2004).

3.2.1. Instalación de Python. Se procedió a instalar Python de la siguiente manera:

1. Fuimos a la página oficial de Python, cuyo link de acceso es: <https://www.python.org/>, ilustrado en la figura 1.

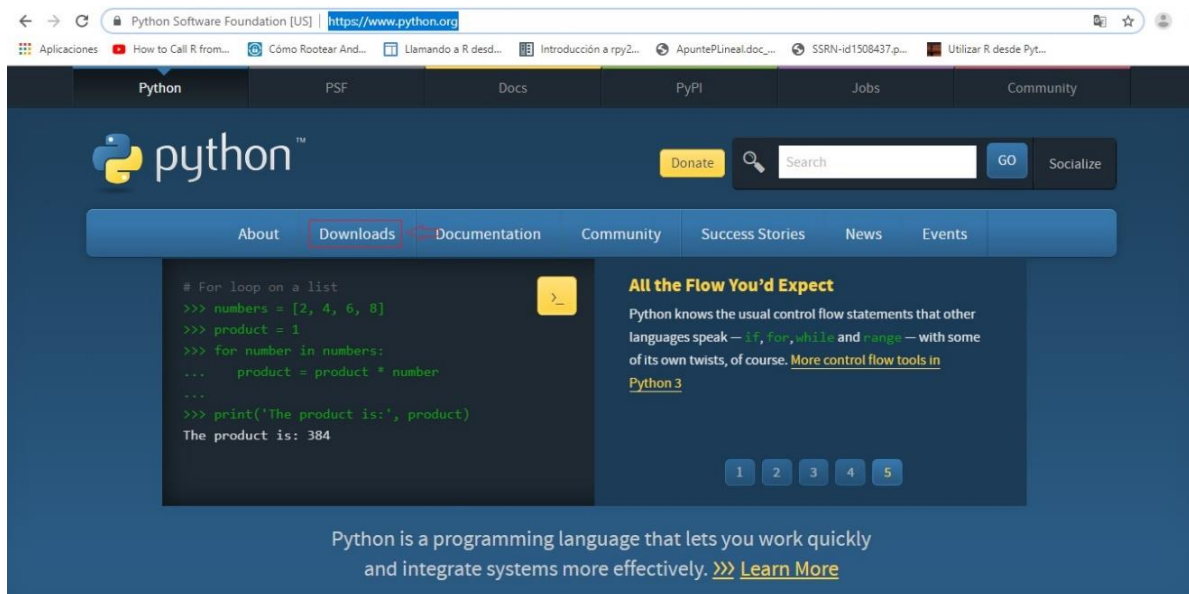


Figura 1. Página oficial de Python

2. Seleccionamos la opción de «Downloads» y seleccionar el sistema operativo de preferencia, en este caso para Windows en su versión más reciente, tal cual como se observa en la figura 2.

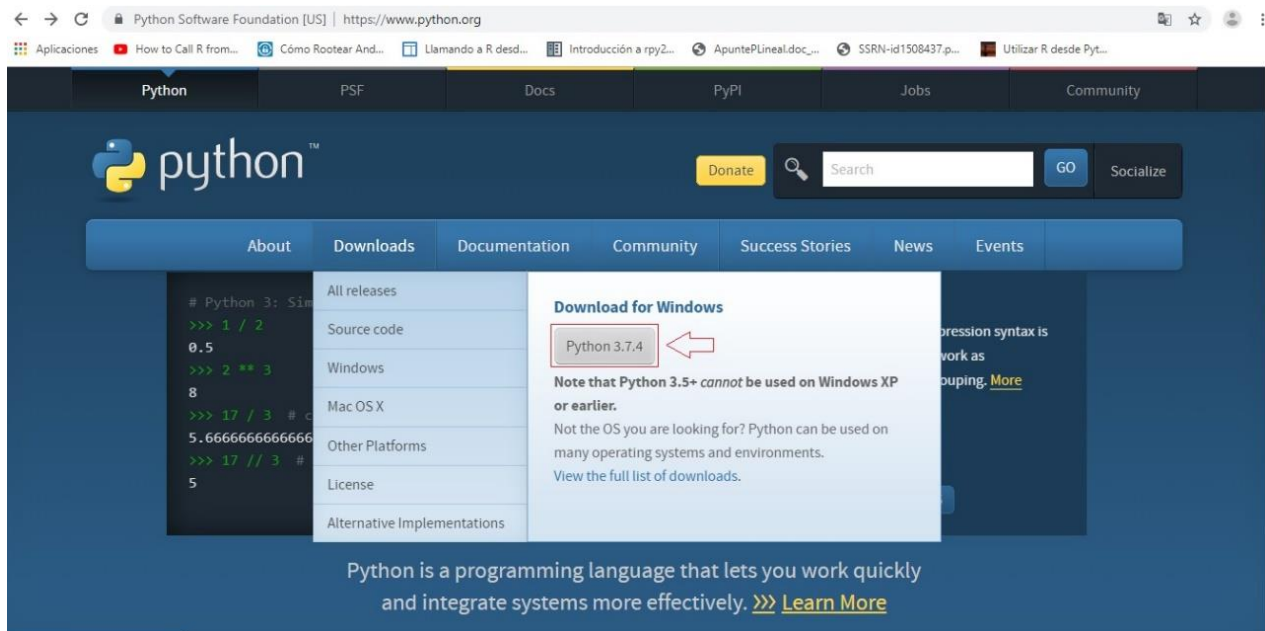


Figura 2. Descarga del instalador de Python

3. Automáticamente se descargará el archivo ejecutable, una vez se ha completado al 100 % se procede a ejecutar como administrador, se recomienda tener presente la ubicación o raíz que define el sistema por defecto para la descarga del lenguaje ya que más adelante será necesaria tenerla presente, luego, seleccionar la opción de «instalar para todos los usuarios» y luego “next o siguiente” hasta llegar al “finish o final”.

4. Es necesario realizar una ligera modificación a las variables de entorno del sistema, esto con el fin de poder utilizar interfaces o entornos de desarrollo integrados especializados que permiten la codificación con el lenguaje de Python, ya que si posiblemente no se hace este paso el sistema no reconozca a Python de manera natural.

- Para eso nos dirigimos al Panel de control
- Sistema y Seguridad
- Configuración avanzada del sistema
- Opciones avanzadas
- Variables de entorno
- Variables del sistema
- Seleccionar la opción «path»
- Seleccionar la opción «Editar», en la figura 3 se puede observar el proceso hasta el momento.

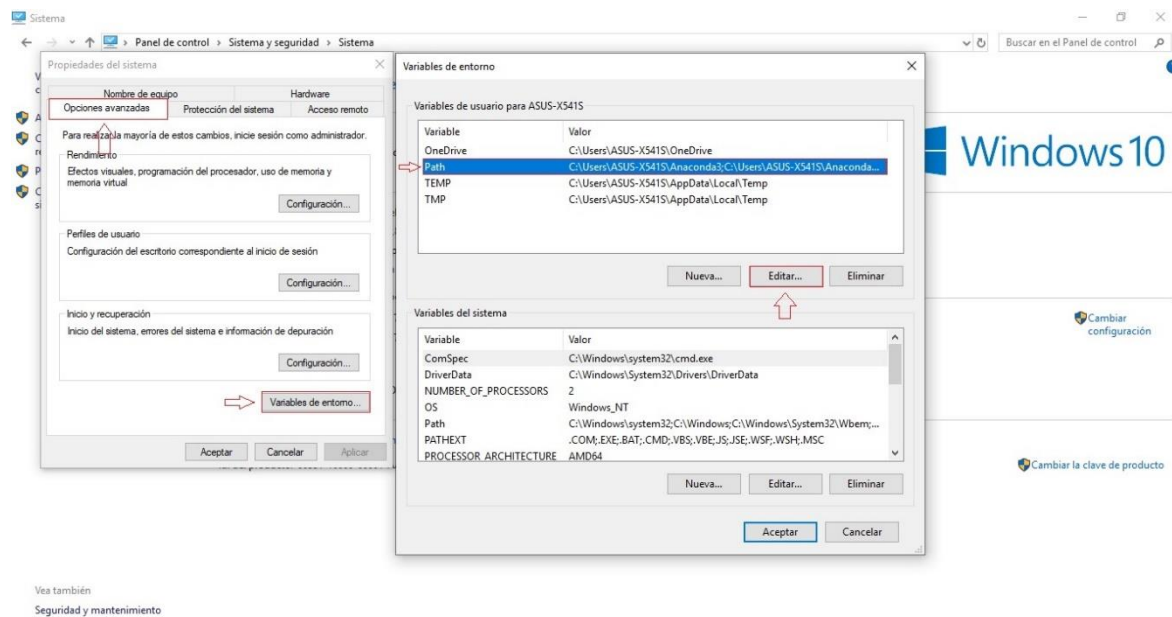


Figura 3. Editar las variables de entorno del sistema

- Seleccionamos la opción de «Nuevo», y allí en el campo que se abre copiar la ubicación del archivo en el que Python se instaló por defecto, tal cual como se observa en la figura 4.

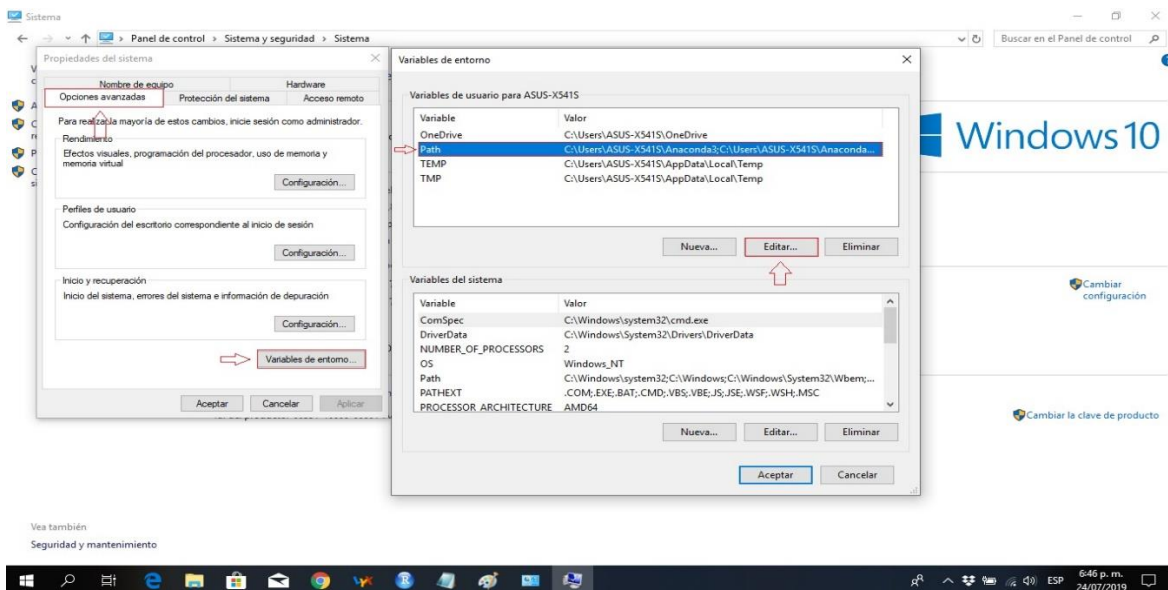
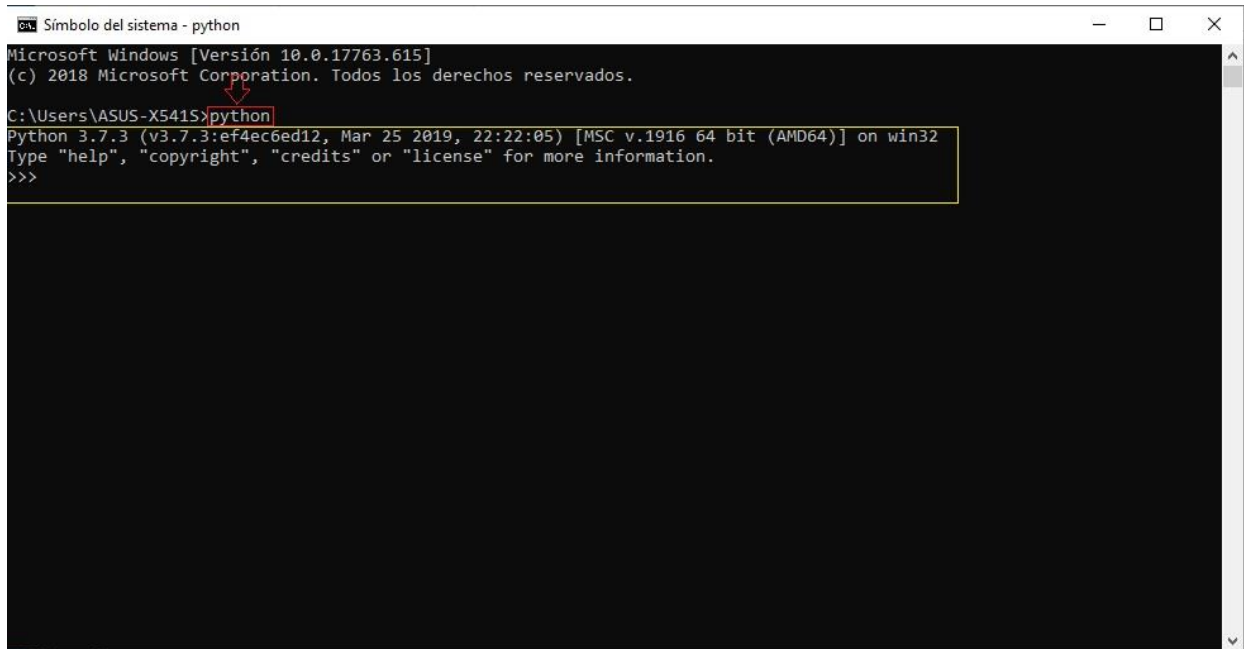


Figura 4. Añadiendo la ubicación de Python al sistema

- Seleccionar la opción de «Aceptar» y listo
- Previamente «Python» ya estaba instalado en nuestro computador, pero para verificar que el proceso fue exitoso, iremos al símbolo del sistema (cmd) a través del buscador de Windows, escribiremos «cmd» + enter, esto me llevara a la ventana por defecto, y allí escribiremos «python», si el sistema reconoce al lenguaje mostrara las características del mismo, tal cual como se observa en la figura 5.



```
Símbolo del sistema - python
Microsoft Windows [Versión 10.0.17763.615]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.
C:\Users\ASUS-X5415>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figura 5. El Sistema reconociendo a Python.

5. La programación a escala profesional utiliza «Entornos de Desarrollo Integrados (IDE), que permiten la facilidad en la escritura de código, para efectos de este código se utilizara Visual Studio Code ya que es un IDE liviano y de fácil aprendizaje (Halvorsen, 2006).

3.2.2. Instalación de R-Project. Se procedió a instalar R Project de la siguiente manera:

1. Se ingresó a la página oficial de R-Project, cuyo link de acceso es: <https://www.r-project.org/>, tal cual como se observa en la figura 6.

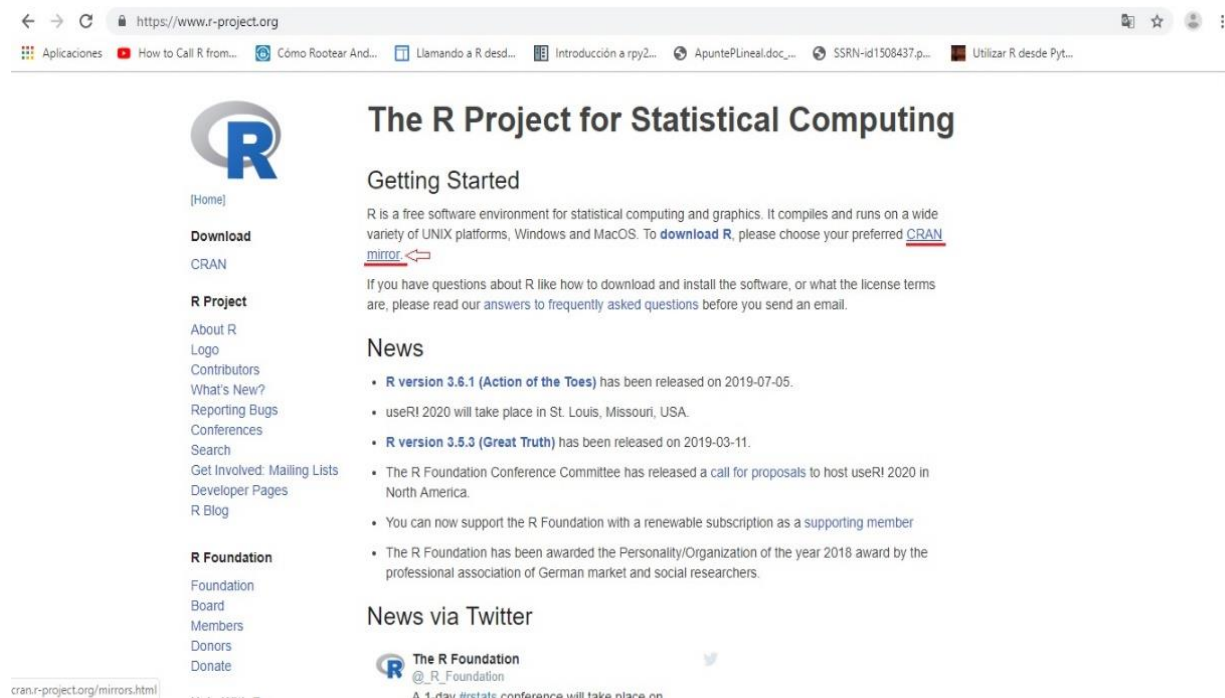


Figura 6. Página oficial de R

2. Seleccionamos la opción descrita en la figura, que posteriormente re-direccionará a la opción de espejo o sitio de distribución de descarga tal cual como se observa en la figura 7.

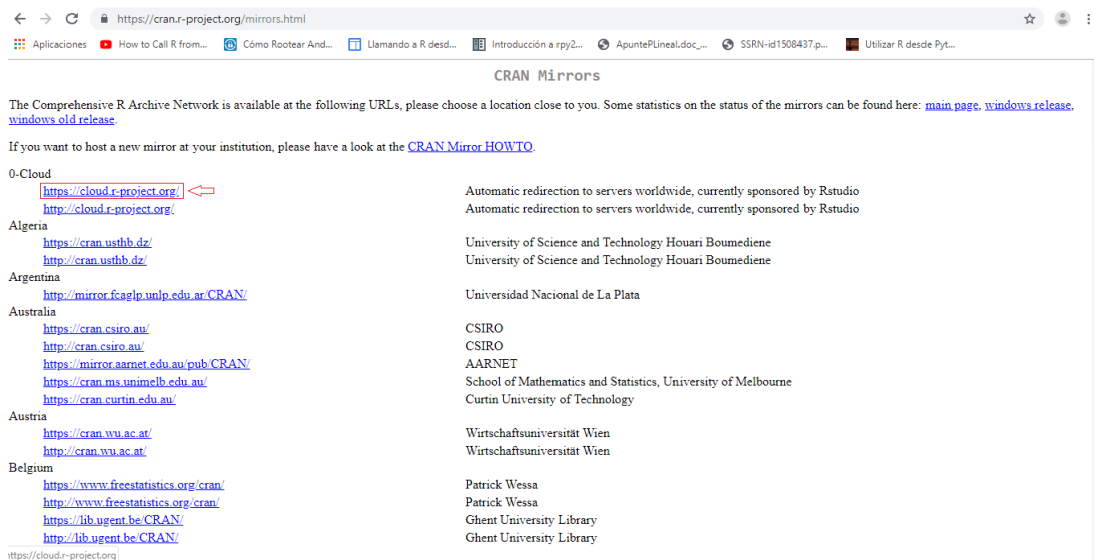


Figura 7. Espejo o sitio de distribución para R-Project

3. Una vez re-direccionado, seleccionamos el sistema operativo de preferencia, en este caso para Windows, y seleccionar la opción de «install R for the first time».
4. Hicimos clic sobre la opción de «Download R form Windows», tal cual como se observa en la figura 8.

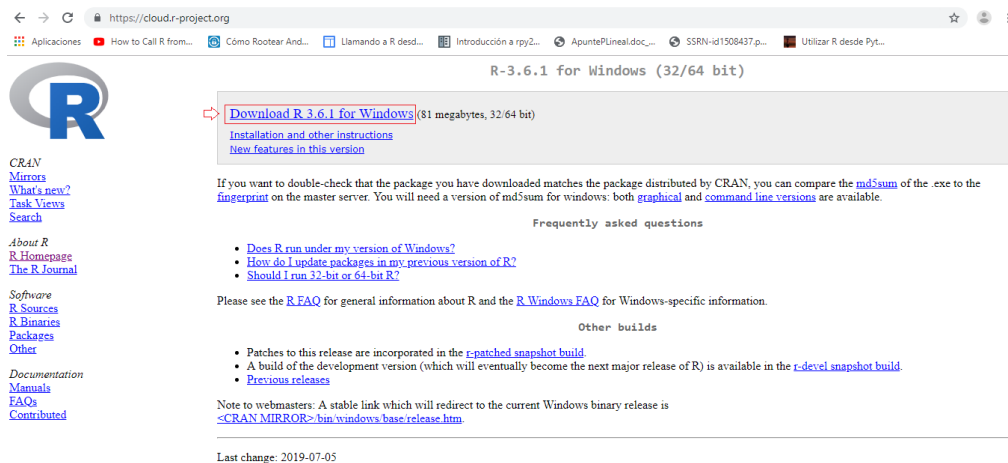


Figura 8. Descarga del instalador de R-Project.

5. Una vez se ha completado la descarga solo queda ejecutar el instalador y esperar hasta completar la instalación.
6. R-Project necesita de una serie de librerías o módulos especializados para llevar a cabo el desempeño normal del lenguaje de programación estadístico, para eso se realizó lo siguiente:
 - Se abrió el entorno de R-Project y seleccionar las opciones descritas, tal cual como se observa en la figura 9.

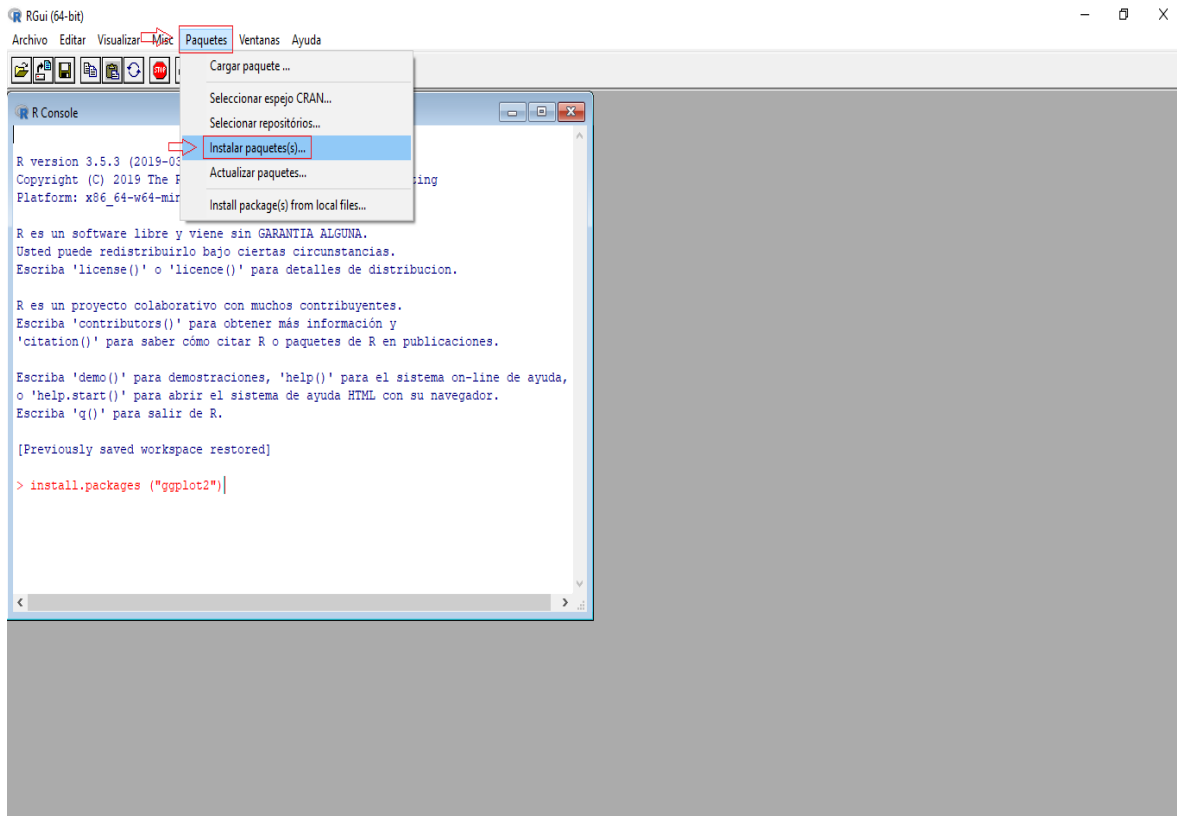


Figura 9. Opción de descarga de las librerías en R-Project.

- Seleccionamos la opción de preferencia deseada, en este caso y por algo más llamado patriotismo seleccionamos a Colombia como el sitio de distribución del lenguaje R-Project, cabe aclarar que entre más cercano nos encontremos al espejo de distribución de software este incide en el ancho de banda, latencia, entre otros (Quora, 2014), tal cual como se observa en la figura 10.

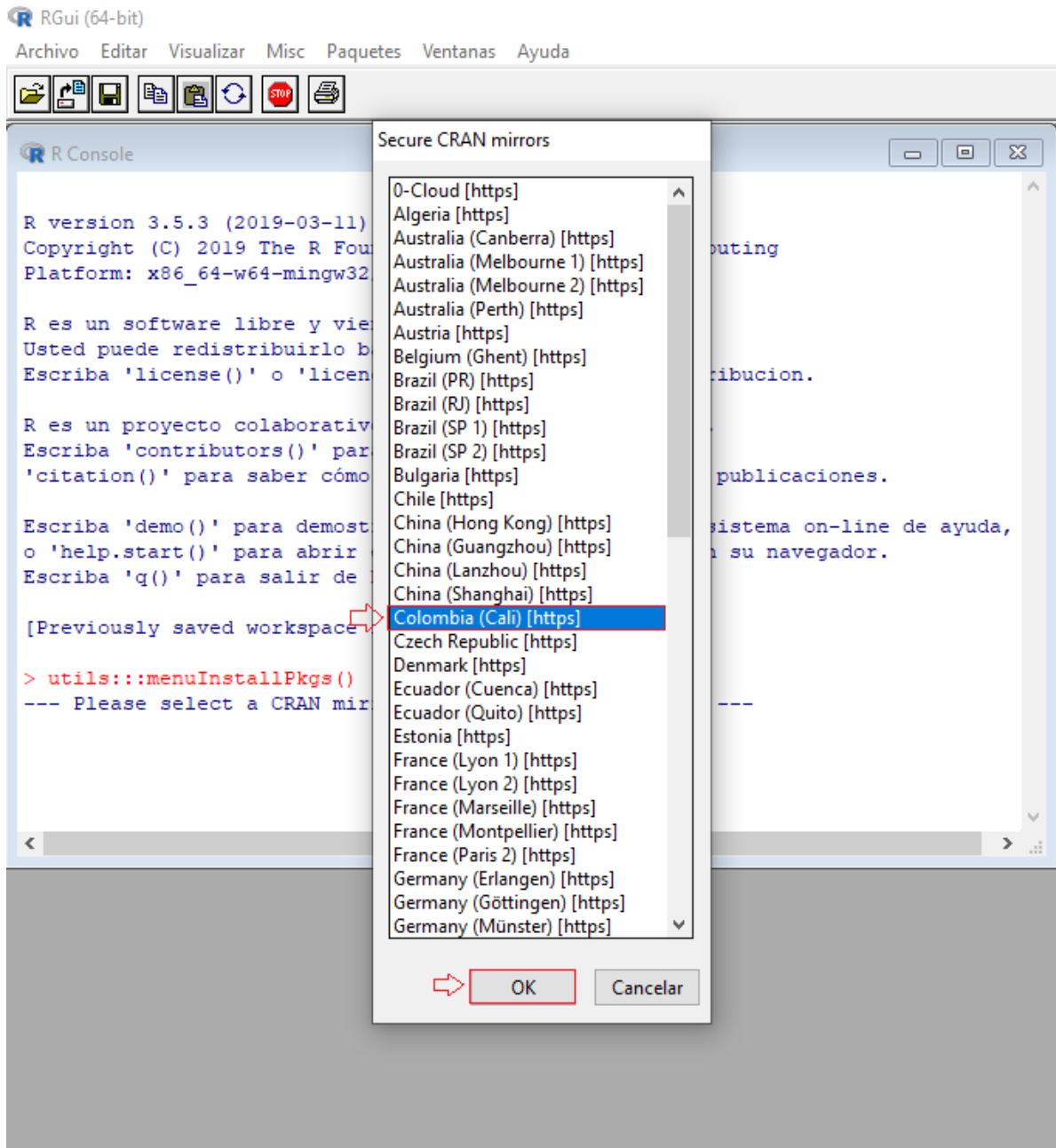


Figura 10. Sitio de distribución de R-Project.

- Se abre una ventana con todos los paquetes o librerías disponibles tal cual como se observa en la figura 11, para efectos académicos se recomienda descargar todas las opciones disponibles, y esperar el tiempo de la descarga que puede oscilar entre las 3 y 6 horas.

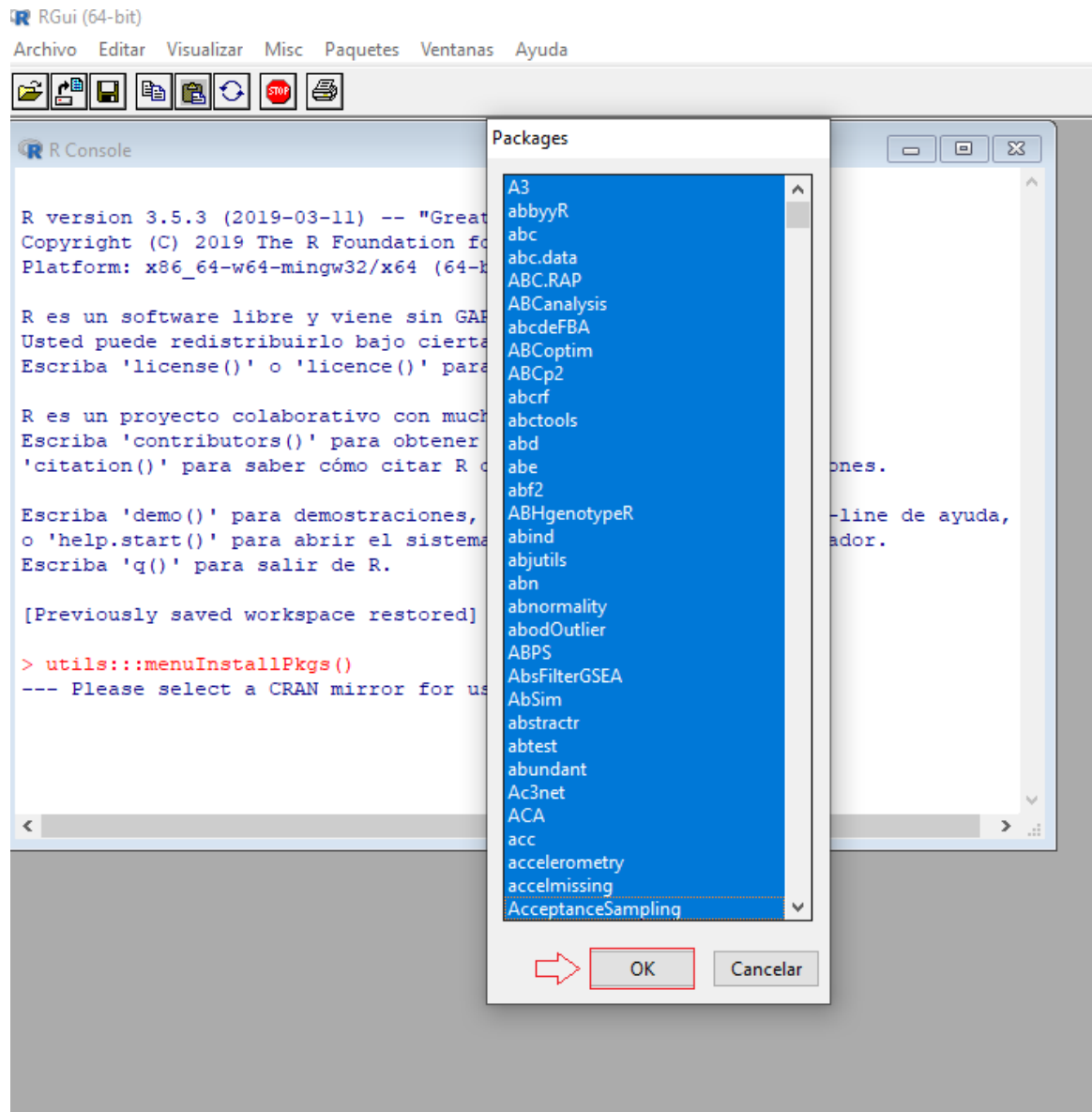


Figura 11. Seleccionando las librerías con las que trabaja R-Project

Aclaración: Si solo se desea trabajar con librerías especializadas, es decir, no se desea tener todas las librerías descargadas, se puede referenciar la librería de preferencia, a través del siguiente código escrito en R-Project (Quora, 2014), en este caso «install.packages («ggplot2»)», de esta forma se digita la función (install.packages) seguido entre paréntesis y entre comillas el nombre de la librería de preferencia, a modo de ejemplo en la figura 12 se instaló la librería «ggplot2» encargada de generar gráficos especializados.

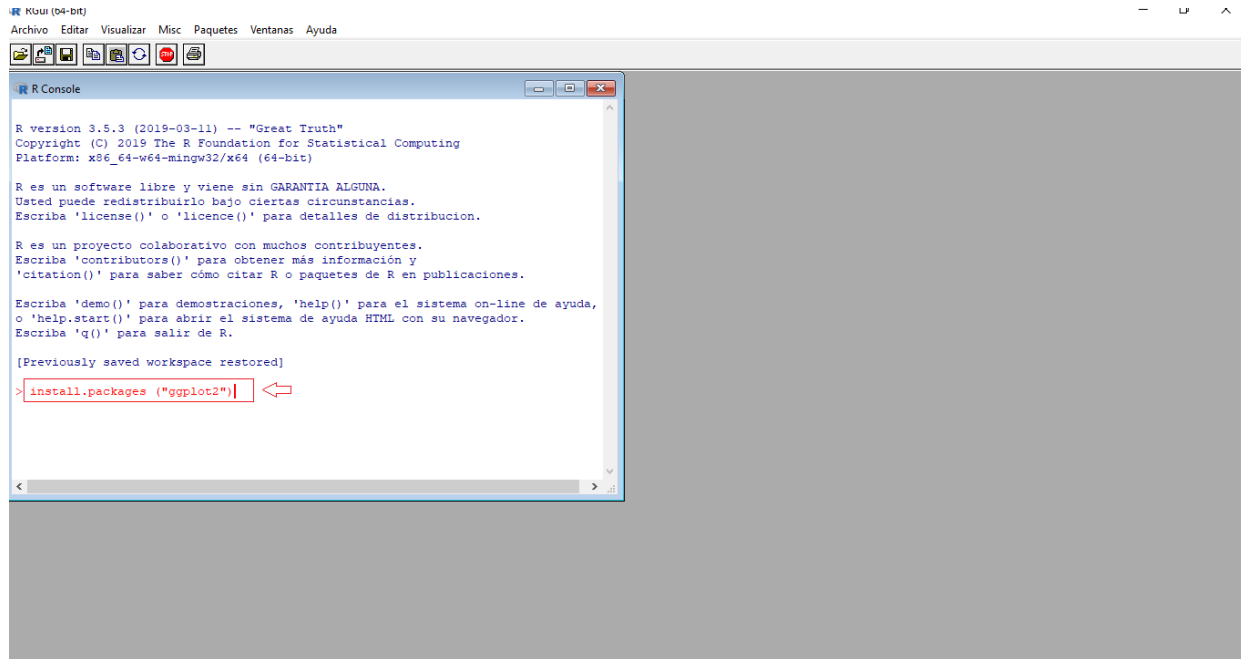


Figura 12. Instalación de paquetes en R-Project a través de código.

- Una vez terminada la descarga de los módulos ya se puede empezar con la programación en R-Project.
- Se recomienda descargar un IDE para trabajar sobre R-Project, ya que como se hacía mención anteriormente estos editores permiten la facilidad en la legibilidad e interpretación del código generado, para efectos de este código se utilizó Tinn R.

3.3. Tabulación y análisis de datos

La información o base de datos forestal fue tabulada en la hoja de cálculo de Excel ya que esta herramienta posee una buena capacidad de almacenamiento y es de las más usadas en el ámbito corporativo de manera constante puesto que en Colombia la mayoría de servidores contienen el sistema operativo de Microsoft Windows (Betancur Calle & Pérez García, 2013) fue necesario realizar una serie de procesos que se describen a continuación:

Nota: Se guardaron los archivos en extensiones de ruta corta, por ejemplo, la base de datos forestal fue guardada en el escritorio, ya que es muy posible que se generen errores si los archivos son guardados en ubicaciones muy extensas a la hora de ser llamados por R-Project.

1. Se tabularon los datos del inventario forestal, destacando las variables como DAP en cm, diámetro sin corteza en cm, altura total del árbol en m y una altura de referencia a 1,30 m tal cual como se observa en la tabla 3.

Tabla 3.

Variables básicas del inventario forestal tabuladas en una hoja de cálculo.

A	B	C	D	E
NOMBRE CIENTIFICO	DAP	DSC	H	Hrod
<i>Eucalyptus globulus</i>	35.00	31.80	19.86	1.30
<i>Eucalyptus globulus</i>	54.50	51.30	28.60	1.30
<i>Eucalyptus globulus</i>	58.20	55.00	31.56	1.30
<i>Eucalyptus globulus</i>	67.50	64.30	33.94	1.30
<i>Eucalyptus globulus</i>	31.60	28.40	17.93	1.30
<i>Eucalyptus globulus</i>	30.80	27.60	18.35	1.30
<i>Eucalyptus globulus</i>	49.70	46.50	31.57	1.30
<i>Eucalyptus globulus</i>	61.90	58.70	32.76	1.30
<i>Eucalyptus globulus</i>	31.00	27.80	25.07	1.30
<i>Eucalyptus globulus</i>	68.00	64.80	30.38	1.30
<i>Eucalyptus globulus</i>	47.00	43.80	33.74	1.30
<i>Eucalyptus globulus</i>	59.60	56.40	34.25	1.30
<i>Eucalyptus globulus</i>	27.80	24.60	24.25	1.30
<i>Eucalyptus globulus</i>	43.30	40.10	34.29	1.30
<i>Eucalyptus globulus</i>	48.00	44.80	28.43	1.30
<i>Eucalyptus globulus</i>	54.30	51.10	29.49	1.30
<i>Eucalyptus globulus</i>	27.30	24.10	14.42	1.30
<i>Eucalyptus globulus</i>	35.90	32.70	28.01	1.30
<i>Eucalyptus globulus</i>	24.60	21.40	12.03	1.30
<i>Eucalyptus globulus</i>	36.80	33.60	23.43	1.30
<i>Eucalyptus globulus</i>	43.80	40.60	23.24	1.30
<i>Eucalyptus globulus</i>	32.50	29.30	34.65	1.30

2. Una vez se ha digitado la información, se verificó la coherencia de los datos rectificando que no se presentaran sesgos sistemáticos, entre otros, además a través del cociente de esbeltez se garantiza encontrar individuos que presenten estructuras regulares (Birchler, Royo, & Pardos, 1998), relaciones que presenten valores elevados deben ser sometidos para su respectivo análisis.

$$CE = \frac{H}{d}$$

Siendo,

H: Altura del individuo

d: Diámetro del individuo

CE: Cociente de esbeltez

3. La literatura recomienda muestreos destructivos para una mayor precisión en el ajuste de los modelos de ahusamiento (Prodan, 1997), sin embargo como no se hizo este procedimiento se tomaron en cuenta las variables anteriormente tabuladas y se describe que a 1 m de la altura total del individuo se encuentra un diámetro de una pulgada, esto debido a que generalmente los muestreos encontrados en la literatura definen que desde este diámetro en adelante a una altura de 1 m o menor no se define un uso comercial (Bustamante, 2014), el proceso es descrito en la tabla 4.

Tabla 4.

Duplicidad de datos para efectuar la convergencia de los parámetros de los modelos

A	B	C	D	E
NOMBRE CIENTIFICO	DAP	DSC	H	Hrod
<i>Eucalyptus globulus</i>	26.5	25.5	19.8644114	1.30
<i>Eucalyptus globulus</i>	35.00	2.54	19.86	18.86
<i>Eucalyptus globulus</i>	58.20	2.54	31.56	30.56
<i>Eucalyptus globulus</i>	67.50	2.54	33.94	32.94
<i>Eucalyptus globulus</i>	31.60	2.54	17.93	16.93
<i>Eucalyptus globulus</i>	30.80	2.54	18.35	17.35
<i>Eucalyptus globulus</i>	49.70	2.54	31.57	30.57
<i>Eucalyptus globulus</i>	61.90	2.54	32.76	31.76
<i>Eucalyptus globulus</i>	31.00	2.54	25.07	24.07
<i>Eucalyptus globulus</i>	68.00	2.54	30.38	29.38
<i>Eucalyptus globulus</i>	47.00	2.54	33.74	32.74
<i>Eucalyptus globulus</i>	59.60	2.54	34.25	33.25
<i>Eucalyptus globulus</i>	27.80	2.54	24.25	23.25
<i>Eucalyptus globulus</i>	48.00	2.54	28.43	27.43
<i>Eucalyptus globulus</i>	54.30	2.54	29.49	28.49
<i>Eucalyptus globulus</i>	35.90	2.54	28.01	27.01
<i>Eucalyptus globulus</i>	24.60	2.54	12.03	11.03
<i>Eucalyptus globulus</i>	36.80	2.54	23.43	22.43
<i>Eucalyptus globulus</i>	43.80	2.54	23.24	22.24
<i>Eucalyptus globulus</i>	32.50	2.54	34.65	33.65
<i>Eucalyptus globulus</i>	35	2.54	27.07791429	26.08
<i>Eucalyptus globulus</i>	35	2.54	37.95516148	36.96

4. A continuación, se seleccionó un 70 % de la base de datos original para realizar el ajuste, y el 30 % restante para validación con esto se garantiza la independencia de los mismos y se evalúa la capacidad predictiva de los modelos (de-Miguel, Mehtätalo, Shater, Kraid, & Pukkala, 2012), éstos datos fueron digitados en la hoja número dos del archivo original, tal cual como se observa en la tabla 5.

Tabla 5.
70 % de los datos para realizar el ajuste

A	B	C	D	E
NOMBRE CIENTIFICO	DAP	DSC	H	Hrod
<i>Eucalyptus globulus</i>	35.00	31.80	19.86	1.30
<i>Eucalyptus globulus</i>	58.20	55.00	31.56	1.30
<i>Eucalyptus globulus</i>	67.50	64.30	33.94	1.30
<i>Eucalyptus globulus</i>	31.60	28.40	17.93	1.30
<i>Eucalyptus globulus</i>	30.80	27.60	18.35	1.30
<i>Eucalyptus globulus</i>	49.70	46.50	31.57	1.30
<i>Eucalyptus globulus</i>	61.90	58.70	32.76	1.30
<i>Eucalyptus globulus</i>	31.00	27.80	25.07	1.30
<i>Eucalyptus globulus</i>	68.00	64.80	30.38	1.30
<i>Eucalyptus globulus</i>	47.00	43.80	33.74	1.30
<i>Eucalyptus globulus</i>	59.60	56.40	34.25	1.30
<i>Eucalyptus globulus</i>	27.80	24.60	24.25	1.30
<i>Eucalyptus globulus</i>	48.00	44.80	28.43	1.30
<i>Eucalyptus globulus</i>	54.30	51.10	29.49	1.30
<i>Eucalyptus globulus</i>	35.90	32.70	28.01	1.30
<i>Eucalyptus globulus</i>	24.60	21.40	12.03	1.30
<i>Eucalyptus globulus</i>	36.80	33.60	23.43	1.30
<i>Eucalyptus globulus</i>	43.80	40.60	23.24	1.30
<i>Eucalyptus globulus</i>	32.50	29.30	34.65	1.30
<i>Eucalyptus globulus</i>	35	34	27.07791429	1.30
<i>Eucalyptus globulus</i>	35	34	37.95516148	1.30
<i>Eucalyptus globulus</i>	34.7	33.7	28.50335288	1.30

▶ Eucalyptus globulus **Ajuste** Validación XTZ +

5. Sobre el 70 % y 30 % de los datos se declararon las variables dependientes e independientes de los modelos ajustados en este código, en este caso (Y, X, T, Z) tal cual como se observa en la tabla 6, cabe aclarar que la nomenclatura descrita en las variables en las hojas de cálculo no obedece a escritura técnica por facilidades de programación.

Tabla 6.

Declaración de las variables dependientes e independientes de los modelos de ahusamiento para su respectivo ajuste

A	B	C	D	E	F	G	H	I
NOMBRE CIENTIFICO	DAP	DSC	H	Hrod	Y	X	T	Z
<i>Eucalyptus globulus</i>	35.00	31.80	19.86	1.30	d^2/dap^2	1	0.065448	0.934552
<i>Eucalyptus globulus</i>	54.50	51.30	28.60	1.30	0.88601633	1	0.045461	0.954539
<i>Eucalyptus globulus</i>	58.20	55.00	31.56	1.30	0.89305747	1	0.041192	0.958808
<i>Eucalyptus globulus</i>	67.50	64.30	33.94	1.30	0.90743265	1	0.038307	0.961693
<i>Eucalyptus globulus</i>	31.60	28.40	17.93	1.30	0.80772312	1	0.072485	0.927515
<i>Eucalyptus globulus</i>	30.80	27.60	18.35	1.30	0.80300219	1	0.070847	0.929153
<i>Eucalyptus globulus</i>	49.70	46.50	31.57	1.30	0.87537296	1	0.041175	0.958825
<i>Eucalyptus globulus</i>	61.90	58.70	32.76	1.30	0.89927994	1	0.039688	0.960312
<i>Eucalyptus globulus</i>	31.00	27.80	25.07	1.30	0.80420395	1	0.051848	0.948152
<i>Eucalyptus globulus</i>	68.00	64.80	30.38	1.30	0.90809689	1	0.042792	0.957208
<i>Eucalyptus globulus</i>	47.00	43.80	33.74	1.30	0.86846537	1	0.038526	0.961474
<i>Eucalyptus globulus</i>	59.60	56.40	34.25	1.30	0.8955002	1	0.037958	0.962042
<i>Eucalyptus globulus</i>	27.80	24.60	24.25	1.30	0.783034	1	0.053598	0.946402
<i>Eucalyptus globulus</i>	43.30	40.10	34.29	1.30	0.85765565	1	0.037916	0.962084
<i>Eucalyptus globulus</i>	48.00	44.80	28.43	1.30	0.87111111	1	0.045727	0.954273
<i>Eucalyptus globulus</i>	54.30	51.10	29.49	1.30	0.88560924	1	0.044088	0.955912
<i>Eucalyptus globulus</i>	27.30	24.10	14.42	1.30	0.77930738	1	0.090169	0.909831
<i>Eucalyptus globulus</i>	35.90	32.70	28.01	1.30	0.82967233	1	0.046408	0.953592
<i>Eucalyptus globulus</i>	24.60	21.40	12.03	1.30	0.75675854	1	0.108052	0.891948
<i>Eucalyptus globulus</i>	36.80	33.60	23.43	1.30	0.83364839	1	0.05548	0.94452
<i>Eucalyptus globulus</i>	43.80	40.60	23.24	1.30	0.85921895	1	0.055941	0.944059
<i>Eucalyptus globulus</i>	32.50	29.30	34.65	1.30	0.8127716	1	0.037521	0.962479

En donde: $Y = \frac{d^2}{dap^2}$; $T = \frac{h}{ht}$; $X = \frac{ht-h}{ht-1.3}$; $Z = \frac{ht-h}{ht}$

Siendo,

d : diámetro sin corteza (cm) a la altura de 1.30 m

DAP: diámetro normal (cm) a la altura de 1.30 m

ht: altura total del árbol (m)

h: altura de referencia del árbol a 1.30 m

3.4. Ajuste de los modelos de ahusamiento

A continuación, se describen los modelos de ahusamiento ilustrados en la tabla 7 ajustados en el desarrollo del script, es necesario aclarar que los modelos utilizados hacen referencia a especies de rápido crecimiento tales como pinos y eucaliptos, esto debido a que dichas especies poseen paquetes tecnológicos bastante desarrollados.

Tabla 7.

Modelos de ahusamiento contenidos en del desarrollo del script

Modelo (P(x))	Autores
$Y = b1 * X^{1.5} + b2 * X^3 + b3 * X^{3.2}$	Bruce, Curtis & Vancoeving (1968)
$Y = b1 * (T - 1) + b2 * (T^2 - 1)$	Kozak, Munro, & Smith (1969)
$Y = b1 * Z (b2)$	Demaerschalk (1972)
$Y = \frac{b1}{dap^2 * h} * Z^{(b2)} + b3 * Z^{(b4)}$	Demaerschalk (1973)
$Y = b1 * X + b2 * X^2 + b3 * X^3 + b4 * X^4 + b5 * X^5$	Lowell (1986)
$Y = b1 * (T - 1) + b2 * \sin(2\pi T) + b3 * \cot(\frac{\pi}{2} T)$	Thomas y Parresol (1991)
$Y = b1 * Z + b2 * Z^2 + b3 + Z^3$	Renteria & Ramirez (1998)
$Y = b1 * X + b2 * X^2 + b3 * X^3$	Coffre (Rojo et al. (2005)
$Y = \frac{X}{b0 + b1 * X + b2 * X^2 + b3 * X^3}$	J. Andrés Rodríguez Toro et al. (2016)

En donde: $Y = \frac{d^2}{dap^2}$; $T = \frac{h}{ht}$; $X = \frac{ht-h}{ht-1.3}$; $Z = \frac{ht-h}{ht}$ siendo,

d: diámetro (cm), de la rodaja obtenida a la altura h (m)

dap: diámetro normal (cm) a la altura de 1.30 m

ht: altura total del árbol (m)

b0, b1, b2, b3, b4, b5: son los parámetros empleados en los modelos.

Se realizó el ajuste de los modelos de ahusamiento mediante la base de datos registrada en la hoja de cálculo anterior, este proceso se hizo a través del lenguaje de programación R-Project (Diéguez-Aranda, 2012), empleando el algoritmo de Gauss Newton (nls) y el algoritmo de Levenbert Marquardt (nlsLM) para la obtención de los parámetros de los modelos, las ecuaciones

que presentan mejor significancia estadística de coeficientes son los empleados para obtener el volumen a través de integración numérica.

3.5. Obtención del volumen

3.5.1. Estimación del volumen a través de integración numérica. Sobre el 30 % de los datos restantes se validará la integración numérica empleando los coeficientes obtenidos del ajuste sobre del 70 % de los datos.

La figura 13 muestra la referencia de la forma geométrica de un árbol.

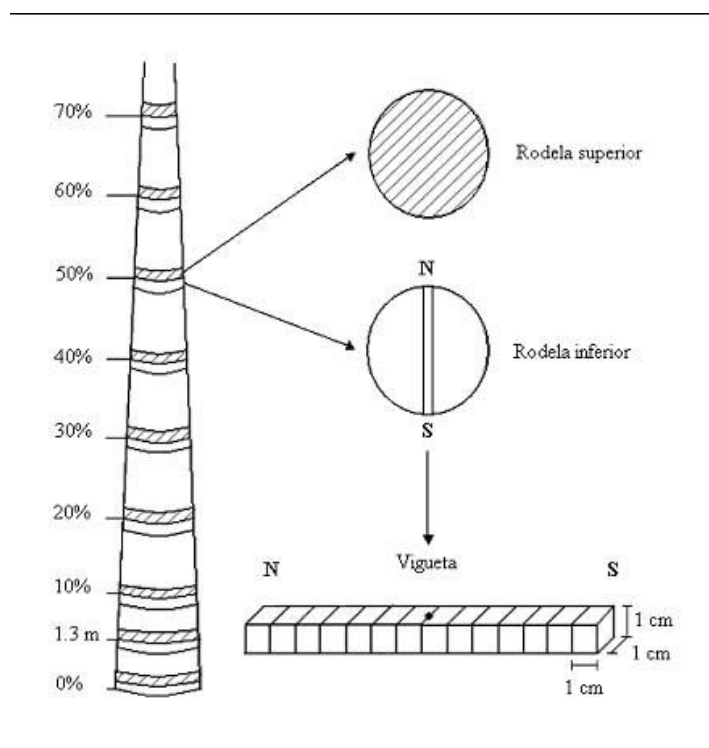


Figura 13. Ejemplo de una forma genérica fustal de un árbol. Adaptado de (Omonte & Valenzuela, 2011).

Para estimar el volumen del fuste, se rotan los modelos de ahusamiento ($P(x)$) sobre su eje en el intervalo $(0, ht)$ (Riaño Melo, 2017), con lo que se obtiene:

$$V = \int_0^{ht} a * DSC^2 * (P(x)) dx$$

En donde,

$a = p/40000 =$ Constante de transformación de diámetros en secciones a m

DSC = diámetro sin corteza a la altura del pecho en (cm) a 1.30 m

$P(x) =$ función de ahusamiento

$0 =$ límite inferior de la integral, este corresponde a la altura del individuo sobre el nivel del suelo (0 m)

$ht =$ límite superior de la integral, este corresponde a la altura total del individuo

En base a la convergencia de los parámetros obtenidos por el ajuste realizado, la estimación del volumen para cada árbol fue calculada de manera numérica en R-Project (Mora, 2016).

3.5.2. Estimación del volumen por método analítico. Mediante este método se asume que la forma del fuste del árbol, para todo el individuo o por secciones, es semejante a sólidos geométricos básicos (cilindro, paraboloides, cono o neiloide) el perfil de dichos sólidos puede representarse mediante la siguiente expresión:

$$y = k * l^r$$

En donde,

y es el radio del sólido en la posición l ,

k es una constante, que define la tasa de aumento del radio del sólido por cada unidad de aumento en la longitud,

r es un exponente que define la forma del sólido (cilindro $r = 0$; paraboloides $r = 0,5$; cono $r = 1$; neiloide $r = 1,5$).

Al girar el área definida entre la curva de perfil y el eje longitudinal, en torno al eje longitudinal, se obtiene un sólido de revolución tal cual como se observa en la figura 14 (Cancino, 2006).

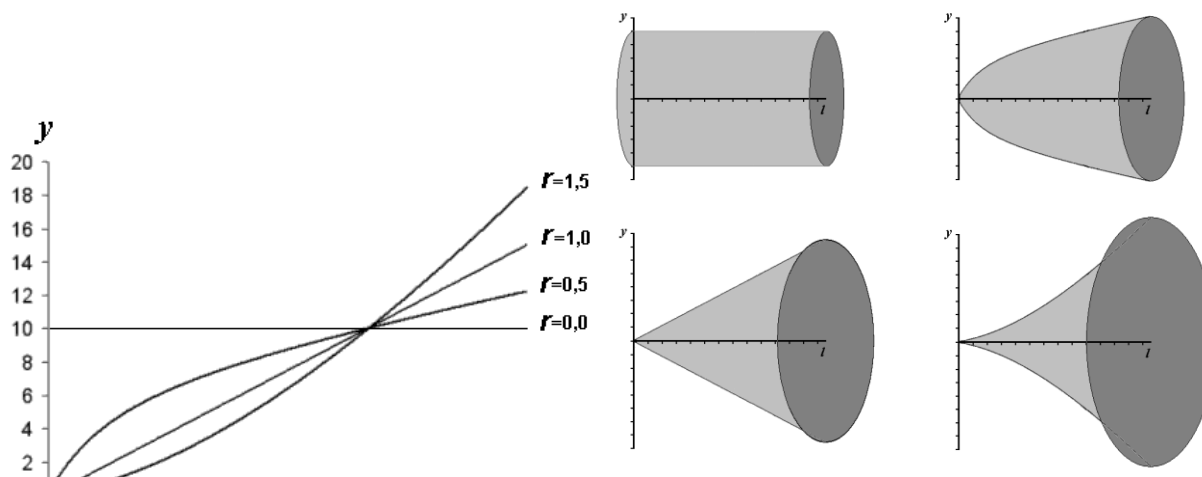


Figura 14. Perfil de sólidos geométricos básicos. Adaptado de (Cancino, 2006).

El volumen de dichos sólidos se obtiene mediante fórmulas específicas, las cuales son utilizadas para la cubicación de árboles y trozas (Cancino, 2006).

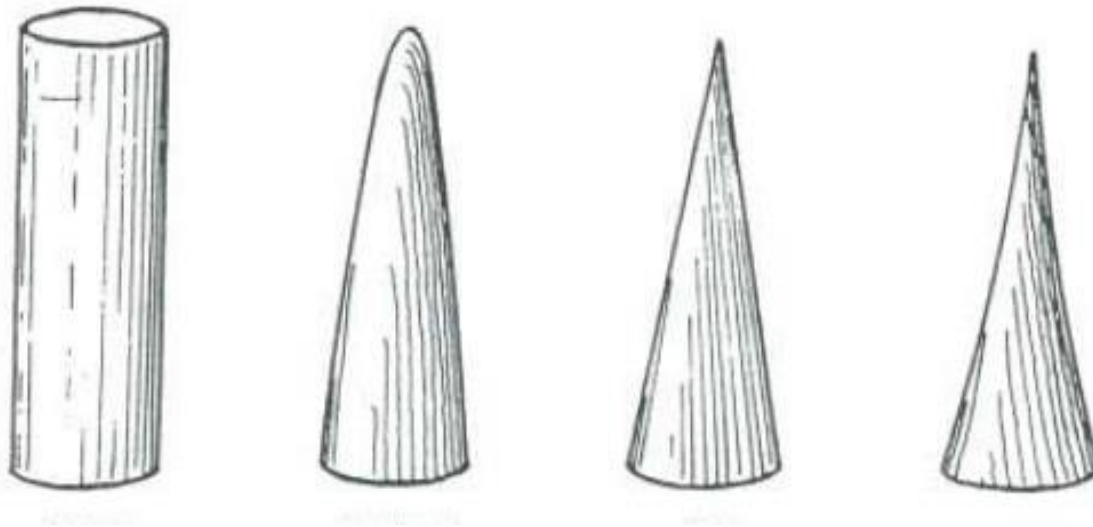


Figura 15. Sólidos de revolución: cilindro, paraboloides, cono y neiloide. Adaptado de (Riaño Melo, 2017).

En la tabla 8 se pueden observar las ecuaciones de los métodos analíticos tradicionales:

Tabla 8.

Ecuaciones de cubicamiento a través de métodos analíticos tradicionales.

Sólido Truncados	Expresión	Fórmula
Paraboloides	$V = \left[\frac{A_M + A_m}{2} \right] L$	Smalian
	$V = A_c L$	Huber
Cono	$V = \left[\frac{A_M + \sqrt[3]{A_M A_m} + A_m}{3} \right] L$	
Neiloide	$V = \left[A_M + \frac{A_M - A_m}{\left(\frac{A_m}{A_M} \right)^3 - 1} \right] \frac{L}{4}$	Northway
Paraboloides, Cono, Neiloide	$V = \left[\frac{A_M + 4A_c + A_m}{6} \right] L$	Newton

A_M , A_m y A_c es el área de la sección mayor, menor y central del sólido truncado respectivamente.

Nota: Expresiones del volumen para sólidos truncados. Adaptado de Estimación de volúmenes de madera en pie de la especie *Pinus patula* mediante polinomio único de ahusamiento. Caso de estudio: finca el guácimo; 2017.

Para la estimación del volumen observado se trató el individuo como un cilindro desde la zona del tocón hasta el DAP, luego a través de la ecuación de Smalian desde la zona del DAP hasta la altura total del individuo como un paraboloides, de tal forma que más adelante se pueda realizar una validación contrastando el valor estimado y el valor observado a través del ajuste de una regresión lineal simple (Rodríguez Toro et al., 2016).

La estimación del volumen por métodos analíticos depende en gran medida de la coherencia entre la forma real de la sección del árbol y la admitida por el sólido de referencia (Prodan, 1997), por esto es necesario identificar el individuo e inmediatamente pensar en el método analítico a calcular, ya sea de manera individual o combinando por secciones.

$$V = (a * D^2 * 1.3) + a * \left(\frac{D^2 + Di^2}{2} \right) * L$$

En donde,

$a = p/40000$ = Constante de transformación de diámetros en secciones a m

D = diámetro (cm) normal a 1.3 (m)

Di = diámetro inferior de la sección (cm)

L = Longitud de la sección fustal (m)

3.6. Bondad de ajuste

Se emplearon los siguientes criterios:

Para comparar los modelos de ahusamiento que presentasen la misma variable respuesta (Y) pero con diferente número de parámetros (tabla 9), se empleó el criterio de información de Akaike (AIC) y el criterio de información Bayesiano (BIC).

Correlación y coeficiente de determinación. Para la selección del mejor modelo en base al volumen se aplicó el análisis de una regresión lineal simple, la correlación existente entre el volumen observado y el volumen predicho se evaluó mediante el coeficiente de determinación (R^2), de tal forma que se pudo validar la confianza del ajuste a través de una interpretación porcentual de credibilidad del resultado obtenido (Sánchez Alberca, 2014).

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

y_i = Valores observados de la variable dependiente (estimación del volumen por método analítico)

\hat{y}_i = Valores del volumen predicho obtenido a través de los modelos de ahusamiento

\bar{y} = Media de los valores observados.

3.6.1. Supuestos básicos de regresión. Para observar el comportamiento de los supuestos de regresión, se empleó el uso de los gráficos estadísticos, de tal forma que se pudiera observar el comportamiento de los modelos que ajustaron de manera significativa.

3.7. Evaluación de la precisión del ajuste del modelo

Para evaluar la precisión del ajuste de los modelos de ahusamiento, se utilizó la raíz del error cuadrático medio (RMSE).

Todos los procedimientos anteriormente especificados fueron corridos en el lenguaje de programación R-Project.

3.8. Interfaz gráfica para obtener el factor de forma de los individuos

En esta línea, la forma del fuste se expresa numéricamente mediante un valor único por árbol.

El factor de forma o mórfico es obtenido de la relación entre el volumen real del árbol y el volumen de un cuerpo geométrico de referencia, en este caso el de un cilindro, estas dimensiones corresponden a las características generales del árbol (DAP, altura). El volumen del sólido de referencia también es denominado como volumen aparente (Cancino, 2006).

$$ff = \frac{vr}{vs}$$

En donde,

ff: factor de forma

vr : volumen del fuste

vs: volumen del solido de referencia o volumen aparente

La interfaz gráfica fue programada en el lenguaje de programación Python tal cual como se observa en la figura 16.

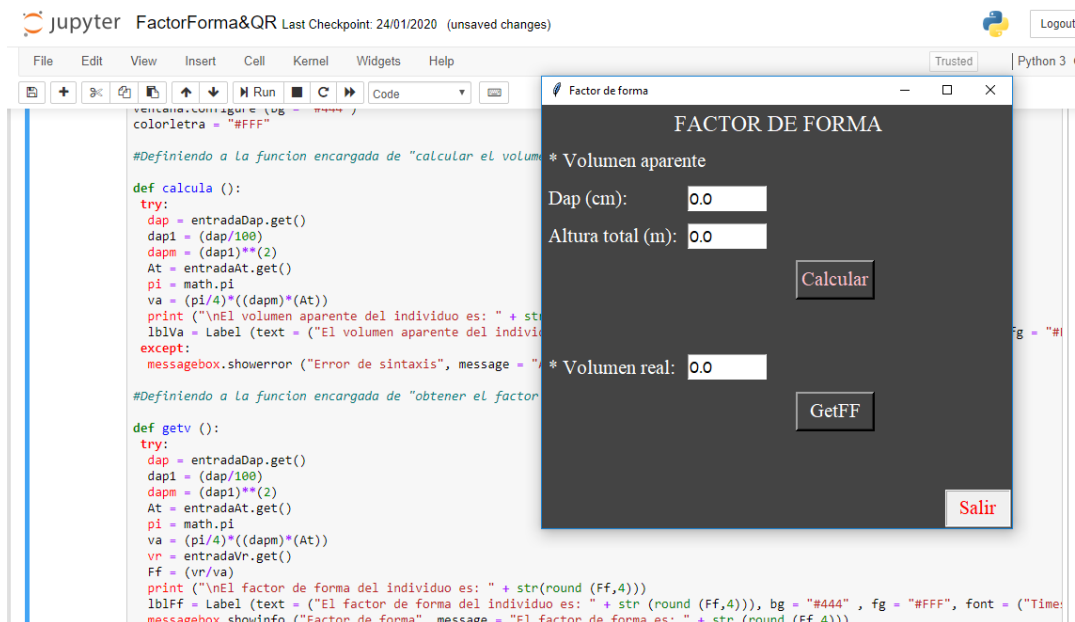


Figura 16. Interfaz gráfica obtenida en Python.

3.9. Código QR

Como una forma de generar una etiqueta digital o física que almacenara la información anteriormente calculada ya sea de un bosque o plantación en específico se genera un código QR a través de Python en formato de imagen .png, con la siguiente información descrita en la tabla 9:

Tabla 9.
Información almacenada en el código QR

Datos almacenados	Datos de entrada
Nombre científico:	
Nombre común:	
Familia:	
Especie:	
Manejo:	
Usos:	
Volumen estimado:	

La información contenida en el script puede ser modificada de acuerdo a las necesidades previstas.

El código fuente está disponible en el siguiente link: <https://unipython.com/generar-codigo-qr-en-python/>

3.10. Formato de etiqueta del código QR

El código QR puede ser leído por una aplicación móvil libre (QR Code RW, entre otras) disponible en las tiendas virtuales que traen los sistemas operativos por defecto, a continuación, se presenta un formato de rotulo previsto para su respectiva lectura, cabe aclarar que el diseño puede ser modificado de acuerdo a las necesidades personales.



Figura 17. Formato de etiqueta código QR.

3.11. Resumen del script generado en R-Project y Python

A continuación, se describe el resumen de los procedimientos realizado por el script generado tanto en R-Project como en Python, con el fin de dar cumplimiento a los objetivos establecidos.

3.11.1. Fase estadística y matemática computacional. Como es descrito en la metodología, el ajuste y la selección de los «mejores modelos» requieren de una serie de tratamientos estadísticos y matemáticos muy importantes, los lenguajes de programación utilizados poseen cómputo estadístico y matemático fuerte, por ende, su respectiva utilización.

3.11.2. Fase en R-Project. A continuación, se describe el procedimiento realizado en R-Project: (Resultado 1)

- Lectura de bases de datos almacenadas en hojas de cálculo (Excel), a través de la librería «readxl».
- Calculo del cociente de esbeltez como un indicador de la regularidad de los individuos.
- Calculo de la cantidad de individuos empleados para la fase de ajuste (70 % de los individuos) y para validación (30 % de los individuos), validación cruzada.
- Estimación de los coeficientes o parámetros de los modelos mediante el algoritmo de Gauss Newton (nls), y el algoritmo de Levenberg Marquardt (nlsLM) para estimaciones que presenten problemas de convergencia.
- Obtención volumétrica a través de integración numérica.
- Obtención volumétrica a través de métodos analíticos.
- Selección del criterio de información de Akaike y Bayesiano.
- Regresión lineal simple (extracción del coeficiente de determinación y gráfico del comportamiento de los supuestos).
- Gráfico del perfil longitudinal fustal.
- Evaluación de la precisión del ajuste del modelo a través de la raíz del error cuadrático medio «RMSE».

- Los procedimientos llevados a cabo en el script buscan que el usuario intente correr las líneas de código, y así pueda darse una idea clara de las fases y/o procedimientos que debe realizar para ajustar modelos de ahusamiento y obtener el volumen de los individuos empleando lenguajes de programación aplicada.

3.11.3. Fase en Python. A continuación, se describe el proceso realizado en Python: (Resultado 2 y 3).

Interfaz gráfica sencilla para obtener el factor de forma para cada individuo a través del módulo «tkinter» (Shipman, 2013).

Generación del código QR que almacena la información anteriormente calculada.

El código generado busca ser un modelo que pueda ser replicado en cálculos no solo de biometría sino en otras áreas de importancia profesional.

3.12. Valor agregado

Es válido afirmar que el desarrollo del script cumple con la información contrastada en la literatura y los objetivos siguen una metodología científica, adicionalmente varias de las líneas de código programadas ya han sido empleadas en otros análisis por estudiantes de la carrera de Ingeniería Forestal de la Universidad Industrial de Santander sede Málaga, que han sido adaptadas a las necesidades de estudio.

4. Resultados

1. A continuación, se presenta el script necesario para llevar a cabo el ajuste de los modelos de ahusamiento en R-Project.

```
#Para resolver problemas de mínimos cuadrados no lineales mediante una
modificación del algoritmo Levenberg-Marquardt se hace uso de la libreria
minpack.lm, esto para problemas de modelos que presentan problemas de
convergencia

#El sistema de R soluciona por defecto el problema de minimos cuadrados no
lineales a traves del algoritmo de Gauss-Newton (nls)

#Para instalar la libreria solo debes ejecutar la siguiente linea de codigo:
#install.packages ("minpack.lm")

#Una vez instalada solo queda importarla si es necesario
#library (minpack.lm)

#-----
-----

#Sobre la base de datos se debe tomar el 70% de los individuos para realizar
el ajuste y el 30% para validacion del modelo, con el fin de evaluar la
capacidad predictiva del mismo.

#Apagar la notacion cientifica
options (scipen = 999)

#-----

#Numero de datos ajustados en los modelos

cat("Digita el total de los individuos: ", ti <- scan ()) #Correr linea

n = ((ti*70)/100)

paste ("El numero de individuos sobre el cual se debe realizar el ajuste es
de: " , trunc(n))
```

```
#-----  
-----  
  
library (readxl)  
tabla <- read_excel("C:/Users/Administrador/Dropbox/proyecto  
grado/DatosEucalyptus.xlsx", sheet=1, col_names=TRUE, col_types=NULL, na="",  
skip=0)  
tabla$esbeltez <- tabla$H/tabla$DAP #Cociente de esbeltez  
head (tabla)  
View (tabla)  
  
#-----  
-----  
  
#Llamando la base de datos a través de la librería readxl sobre el 70% del  
total de los individuos (haciendo el ajuste), hoja numero dos...  
  
library (readxl)  
tabla2 <- read_excel("C:/Users/Administrador/Dropbox/proyecto  
grado/DatosEucalyptus.xlsx", sheet=2, col_names=TRUE, col_types=NULL, na="",  
skip=0)  
head (tabla2)  
View (tabla2)  
  
#-----  
-----  
  
#Las variables invocadas de la base de datos en la hoja de calculo deben  
tener la misma escritura, por ejemplo, si deseo llamar a la variable Y en R  
esta debe estar declarada de la misma manera ya sea (y minúscula o Y
```

MAYUSCULA), de esta manera no se generaran posibles errores por objetos no encontrados la base de datos original

```
#-----  
-----
```

```
#Ajustando los modelos
```

```
#-----  
-----
```

```
##-----  
-----
```

```
#Modelo de ahusamiento propuesto por Bruce, Curtis & Vancoevering (1968)
```

```
##-----  
-----
```

```
RnlinealB <- Y ~ ((b1*X^1.5) + (b2*X^3) + (b3*X^32))
```

```
nlmodB <- nls (RnlinealB, data = tabla2, start = list(b1 = 0.1, b2 = 0.1, b3  
= 0.1))
```

```
summary (nlmodB)
```

```
coe <- coef (nlmodB)
```

```
b1 = coe [1]
```

```
b1
```

```
b2 = coe [2]
```

```
b2
```

```
b3 = coe [3]
```

```
b3
```

```
#-----  
-----  
  
#Modelo de ahusamiento propuesto por Kozak, Munro, & Smith (1969)  
  
##-----  
-----  
  
RnlinealK <- Y ~ (b1*(T - 1) + b2*(T^2 - 1))  
nlmodK <- nls (RnlinealK, data = tabla2, start = list(b1 = 0.1, b2 = 0.1))  
summary (nlmodK)  
coe <- coef (nlmodK)  
b1 = coe [1]  
b1  
b2 = coe [2]  
b2  
  
#Ajustando el volumen sobre el 30% de los datos, fase de validacion, hoja  
numero tres, sheet=3  
  
library (readxl)  
tabla3 <- read_excel("C:/Users/Administrador/Dropbox/proyecto  
grado/DatosEucalyptus.xlsx", sheet=3, col_names=TRUE, col_types=NULL, na="",  
skip=0)  
head (tabla3)  
View (tabla3)  
  
#Integrar modelo de Kozak para todos los individuos
```

```

#Correr todo el ciclo for
for (i in 1:nrow (tabla3)){
  dsc <- tabla3$DSC [i]
  ht <- tabla3$H [i]
  a = pi/40000
  k <- a*(dsc^2)
  fk <- function(h) k*(b1*((h/ht)-1)+b2*((h/ht)^2-1))
  volk <- integrate (fk,0,ht)
  print(volk$value)
}

#Estimacion del volumen por metodos analiticos

# El metodo analitico tomara como forma del arbol un cilindro desde la zona
del tocon hasta el diametro a la altura del pecho, y un paraboloides (Smalian)
desde el dap hasta la altura total del individuo.

#Volumenes (Observados vs predichos), sobre el 30% de los datos, hoja numero
3

a = pi/40000
di = 0.01 #tomado como diámetro inferior de aproximadamente 1 cm
l = (tabla3$H-1.3) #longitud fustal para estimacion del paraboloides
l
tabla3$vol_metanalitico <- (pi/40000 * ((tabla3$DSC)^2) * 1.3) +
(a*((tabla3$DSC^2+di^2)/2)*l)
vobs <- tabla3$vol_metanalitico
vobs
View (tabla3)

```

```
#-----  
-----  
  
#Coeficiente de determinación, correlación entre el volumen observado (Método  
analítico) y volumen predicho (volumen ajustado a través de los modelos de  
ahusamiento)  
  
#-----  
-----  
  
rls = lm (volkozak ~ vobs, data = tabla3)  
summary(rls)#muestra las características estadísticas del ajuste del modelo  
coef(rls)#evaluar los coef del modelo  
residuals (rls) #Invocando los residuales del modelo de regresión  
  
# Obteniendo el coeficiente de determinación, los valores mas cercanos al 1  
demostrarán una mayor aptitud del modelo  
  
rlsc = summary (rls)  
r2 = rlsc [8]  
r2  
  
#Graficos para observar el cumplimiento de los supuestos:  
  
#1. Aleatoriedad de los residuales  
#2. Distribucion normal  
#3. Homocedasticidad de varianza  
#4. Los residuos tienen varianza constante
```

```
win.graph()

par (mfrow = c(2,2)) #Muestra en una ventana el grafico de los supuestos
plot (lm(volkozak ~ vobs, data = tabla3))
```

```
win.graph()

par (mfrow = c(2,2))

rls <- lm(volkozak ~ vobs, data = tabla3)

plot(residuals(rls) ~ tabla3$volkozak, xlab = "Volumen Predicho", ylab =
"Residuales", main = "Aleatoriedad de los residuales", col = "blue")

abline(h = 0, lty = 2)
```

```
#Cálculo de la raíz del error cuadrático medio RMSE
```

```
#n = número de datos ajustados en el modelo
```

```
RMSE = ((sum(residuals(rls)))/(n-2))^1/2
paste ("La raíz del error cuadrático medio es: ", trunc(RMSE))
```

```
##-----
-----
```

```
#Modelo de ahusamiento propuesto por Demaerschalk (1972)
```

```
##-----
-----
```

```
RnlinealD72 <- Y ~ (b1 * (Z^b2))

nlmodD72 <- nls (RnlinealD72, data = tabla2, start = list(b1 = 0.1, b2 =
0.1))
```

```
summary (nlmodD72)

coe <- coef (nlmodD72)

b1 = coe [1]

b1

b2 = coe [2]

b2

#Ajustando el volumen sobre el 30% de los datos, fase de validacion, hoja
numero tres, sheet=3

library (readxl)

tabla3 <- read_excel("C:/Users/Administrador/Dropbox/proyecto
grado/DatosEucalyptus.xlsx", sheet=3, col_names=TRUE, col_types=NULL, na="",
skip=0)

head (tabla3)

View (tabla3)

#Integrar modelo de Demaerschalk (1972) para todos los individuos
#Correr todo el ciclo for
for (i in 1:nrow (tabla3)){
  dsc <- tabla3$DSC [i]
  ht <- tabla3$H [i]
  a = pi/40000
  k <- a*(dsc^2)
  fd72 = function(h) k*(b1*((ht-h)/(ht))^b2)
  voldemaerschalk <- integrate (fd72,0,ht)
  print(voldemaerschalk$value)
}
```

```
#Estimacion del volumen por metodos analiticos

#El metodo analitico tomara como forma del arbol un cilindro desde la zona
del tocon hasta el diametro a la altura del pecho, y un paraboloides (Smalian)
desde el dap hasta la altura total del individuo.

#Volumenes (Observados vs predichos), sobre el 30% de los datos, hoja numero
3

a = pi/40000
di = 0.01 #tomado como diámetro inferior de aproximadamente 1 cm
l = (tabla3$H-1.3) #longitud fustal para estimacion del paraboloides
l
tabla3$vol_metanalitico <- (pi/40000 * ((tabla3$DSC)^2) * 1.3) +
(a*((tabla3$DSC^2+di^2)/2)*l)
vobs <- tabla3$vol_metanalitico
vobs
View (tabla3)

#-----
-----

#Regresion lineal simple, entre el volumen observado (Metodo analitico) y
volumen predicho (volumen ajustado a traves de los modelos de ahusamiento)

#-----
-----

rls = lm (voldemaerschalk ~ vobs, data = tabla3)
```

```
summary(rls)#muestra las características estadísticas del ajuste del modelo
coef(rls)#evaluar los coef del modelo
residuals (rls) #Invocando los residuales del modelo de regresion

# Obteniendo el coeficiente de determinacion, los valores mas cercanos al 1
demostrarán una mayor aptitud del modelo

rlsc = summary (rls)
r2 = rlsc [8]
r2

#Graficos para observar el cumplimiento de los supuestos:

#1. Aleatoriedad de los residuales
#2. Distribución normal
#3. Homocedasticidad de varianza
#4. Los residuos tienen varianza constante

win.graph()
par (mfrow = c(2,2)) #Muestra en una ventana el grafico de los supuestos
plot (lm(voldemaerschalk ~ vobs, data = tabla3))

win.graph()
par (mfrow = c(2,2))
rls <- lm(voldemaerschalk ~ vobs, data = tabla3)
plot(residuals(rls) ~ tabla3$voldemaerschalk, xlab = "Volumen Predicho", ylab
= "Residuales", main = "Aleatoriedad de los residuales", col = "blue")
abline(h = 0, lty = 2)
```

```
#Calculo de la raiz del error cuadratico medio RMSE

#n = numero de datos ajustados en el modelo
RMSE = ((sum(residuals(rls)))/(n-2))^1/2
paste ("La raiz del error cuadratico medio es: ", trunc(RMSE))

##-----
-----

#Modelo de ahusamiento propuesto por Demaerschalk (1973)

##-----
-----

RnlinealD73 <- Y ~ ((b1 / DSC^2 * H) * Z^b2 + b3 * Z^b4)
nlmodD73 <- nls (RnlinealD73, data = tabla2, start = list(b1 = 0.1, b2 = 0.1,
b3 = 0.1, b4 = 0.1))
summary (nlmodD73)
coe <- coef (nlmodD73)

b1 = coe [1]
b1
b2 = coe [2]
b2
b3 = coe [3]
b3
b4 = coe [4]
b4

##-----
-----
```

```
#Modelo de ahusamiento propuesto por Lowell (1986)
```

```
##-----  
-----
```

```
RnlinealL <- Y ~ ((b1*X) + (b2*X^2) + (b3*X^3) + (b4*X^4) + (b5*X^5))  
nlmodL <- nls (RnlinealL, data = tabla2, start = list(b1 = 0.1, b2 = 0.1, b3  
= 0.1, b4 = 0.1, b5 = 0.1))  
summary (nlmodL)  
coe <- coef (nlmodL)  
b1 = coe [1]  
b1  
b2 = coe [2]  
b2  
b3 = coe [3]  
b3  
b4 = coe [4]  
b4  
b5 = coe [5]  
b5
```

```
##-----  
-----
```

```
#Modelo de ahusamiento propuesto por Thomas y Parresol (1991)
```

```
##-----  
-----
```

```
#Argumentos expresados en radianes
```

```
RtTP <- Y ~ ((b1*T-1) + b2*sin(2*pi*T) + b3*(1/tan(pi/2*T)))
```

```
modtTP <- nls (RtTP, data = tabla2, start = list (b1 = 0.1, b2 = 0.1, b3 =  
0.1))
```

```
summary (modtTP)
```

```
coe <- coef (modtTP)
```

```
b1 = coe [1]
```

```
b1
```

```
b2 = coe [2]
```

```
b2
```

```
b3 = coe [3]
```

```
b3
```

```
##-----  
-----
```

```
#Modelo de ahusamiento propuesto por Renteria & Ramirez (1998)
```

```
##-----  
-----
```

```
RnlinealR <- Y ~ ((b1*Z) + (b2*Z^2) + (b3*Z^3))
```

```
nlmodR <- nls (RnlinealR, data = tabla2, start = list(b1 = 0.1, b2 = 0.1,  
b3=0.1))
```

```
summary (nlmodR)
```

```
coe <- coef (nlmodR)
```

```
b1 = coe [1]
```

```
b1
```

```
b2 = coe [2]
```

```
b2
```

```
b3 = coe [3]
```

```
b3
```

```
##-----  
-----
```

```
#Modelo de ahusamiento propuesto por Coffre (Rojo et al. (2005)
```

```
##-----  
-----
```

```
RnlinealC <- Y ~ ((b1*X) + (b2*X^2) + (b3*X^3))
```

```
nlmodC <- nls (RnlinealC, data = tabla2, start = list(b1 = 0.1, b2 = 0.1, b3  
= 0.1))
```

```
summary (nlmodC)
```

```
coe <- coef (nlmodC)
```

```
b1 = coe [1]
```

```
b1
```

```
b2 = coe [2]
```

```
b2
```

```
b3 = coe [3]
```

```
b3
```

```
##-----  
-----
```

```
#Modelo de ahusamiento propuesto por J. Andres Rodriguez Toro et al (2015)
```

```
##-----  
-----  
  
#A través de la función 1.nls (modelo no lineal por mínimos cuadrados) y  
2.nlsLM (una versión modificada de nls que usa nls.lm para realizar el ajuste  
por el método de convergencia de Levenberg-Marquardt)  
  
#1  
  
RnlinealT <- Y ~ X/(b0+b1*X+b2*X^2+b3*X^3)  
  
nlmodT <- nls (RnlinealT, data = tabla2, start = list(b0 = 0.1, b1 = 0.1, b2  
= 0.1, b3 = 0.1))  
  
summary (nlmodT)#Muestra el valor de los coeficientes calculados  
  
coe <- coef (nlmodT)  
  
b0 = coe [1]  
  
b0  
  
b1 = coe [2]  
  
b1  
  
b2 = coe [3]  
  
b2  
  
b3 = coe [4]  
  
b3  
  
  
#2 #install.packages("minpack.lm")  
  
library (minpack.lm)  
  
  
RnlinealT <- Y ~ X/(b0 + b1*X + b2*X^2 + b3*X^3)  
  
LMT <- nlsLM (RnlinealT, data = tabla2, start = list(b0 = 0.1, b1 = 0.1, b2 =  
0.1, b3 = 0.1))
```

```
summary (LMT)

coe <- coef (LMT)

b0 = coe [1]

b0

b1 = coe [2]

b1

b2 = coe [3]

b2

b3 = coe [4]

b3

#-----

-----

#Bondad de ajuste

#Indice de Criterio Akaike

#Al comparar modelos ajustados por la máxima probabilidad a los mismos datos,
cuanto más pequeño sea el AIC o BIC, mejor será el ajuste.

#?AIC

IAB = AIC (nlmodB)

IAK = AIC (nlmodK)

IAD72 = AIC (nlmodD72)
```

```
IAD73 = AIC (LMD)
```

```
IAL = AIC (nlmodL)
```

```
IATP = AIC (modtTP)
```

```
IAR = AIC (nlmodR)
```

```
IAC = AIC (nlmodC)
```

```
IAT = AIC (nlmodT)
```

```
#Estructura de control para seleccion del criterio para Akaike en base al  
ajuste de los modelos
```

```
if ((IAR)<(IAK) && (IAR)<(IAB) && (IAR)<(IAD72) && (IAR)<(IATP)) {  
  print ("El modelo mejor ajustado por el indice de Akaike es el de:  
Renteria")  
} else if ((IAK)<(IAR) && (IAK)<(IAB) && (IAK)<(IAD72) && (IAK)<(IATP)) {  
  print ("El modelo mejor ajustado por el indice de Akaike es el de: Kozak")  
} else if ((IAB)<(IAR) && (IAB)<(IAK) && (IAB)<(IAD72) && (IAB)<(IATP)) {  
  print ("El modelo mejor ajustado por el indice de Akaike es el de: Bruce")  
} else if ((IAD72)<(IAR) && (IAD72)<(IAK) && (IAD72)<(IAB) && (IAD72)<(IATP))  
{  
  print ("El modelo mejor ajustado por el indice de Akaike es el de:  
Demaerschalk (72)")  
} else if ((IATP)<(IAR) && (IATP)<(IAK) && (IATP)<(IAB) && (IATP)<(IAD72)) {  
  print ("El modelo mejor ajustado por el indice de Akaike es el de: Thomas  
Parresol")  
}
```

```
}
```

```
#-----
```

```
-----
```

```
#indice de Criterio Bayesiano
```

```
#?BIC
```

```
IBB = BIC (nlmodB)
```

```
IBK = BIC (nlmodK)
```

```
IBD72 = BIC (nlmodD72)
```

```
IBD73 = BIC (LMD)
```

```
IBL = BIC (nlmodL)
```

```
IBTP = BIC (modtTP)
```

```
IBR = BIC (nlmodR)
```

```
IBC = BIC (nlmodC)
```

```
IBT = BIC (nlmodT)
```

```
#Estructura de control para seleccion del criterio para Bayesiano
```

```
if ((IBR)<(IBK) && (IBR)<(IBB) && (IBR)<(IBD72) && (IBR)<(IBTP)) {
    print ("El modelo mejor ajustado por el indice de Bayesiano es el de:
Renteria")
} else if ((IBK)<(IBR) && (IBK)<(IBB) && (IBK)<(IBD72) && (IBR)<(IBTP)) {
    print ("El modelo mejor ajustado por el indice de Bayesiano es el de:
Kozak")
} else if ((IBB)<(IBR) && (IBB)<(IBK) && (IBB)<(IBD72)&& (IBR)<(IBTP)) {
    print ("El modelo mejor ajustado por el indice de Bayesiano es el de:
Bruce")
} else if ((IBD72)<(IBR) && (IBD72)<(IBK) && (IBD72)<(IBB)&& (IBR)<(IBTP)) {
    print ("El modelo mejor ajustado por el indice de Bayesiano es el de:
Demaerschalk (72)")
} else if ((IBTP)<(IBR) && (IBTP)<(IBK) && (IBTP)<(IBB) && (IBTP)<(IBD72)) {
    print ("El modelo mejor ajustado por el indice de Bayesiano es el de:
Thomas y Parresol")
}
#-----
-----
#-----
-----

#Graficando los perfiles longitudinales fustales

#Que modelos se ajustaron de manera significativa? (En este caso)
#(En este caso):
#1.Demaerschalk (72)
#2.Kozak
#3.Thomas y Parresol
```

```
#Crear un vector llamado XTZ, con valores que van desde 0 hasta 1 de 0.01 en
0.01
```

```
XTZ <- seq(from = 0, to = 1, by = 0.01) print(XTZ)
```

```
#1.Reemplazar los coeficientes obtenidos de acuerdo al modelo ajustado (En
este caso KOZAK) y encontrar la variable respuesta (y)
```

```
summary (nlmodK)
```

```
coe <- coef (nlmodK)
```

```
b1 <- coe [1]
```

```
b1
```

```
b2 <- coe [2]
```

```
b2
```

```
y <- (b1*(XTZ - 1) + b2*(XTZ^2 - 1))
```

```
print(y)
```

```
win.graph ()
```

```
k = plot (XTZ ~ y, data = tabla2, type = "o", lwd = 2, main = "Perfil de
Ahusamiento Kozak", xlab = "XTZ", ylab = "Y", col = "blue") grid (col =
"black")
```

```
#2.Reemplazar los coeficientes obtenidos de acuerdo al modelo ajustado (En
este caso Demaerschalk (72)) y encontrar la variable respuesta (y)
```

```
summary (nlmodD72)
```

```
coe <- coef (nlmodD72)
```

```
b1 <- coe [1]
```

```
b1
```

```
b2 <- coe [2]

b2

y <- (b1 * (XTZ^b2))

print (y)

win.graph ()

d = plot (XTZ ~ y, data = tabla2, type = "o", lwd = 2, main = "Perfil de
Ahusamiento Demaerschalk (72)", xlab = "XTZ", ylab = "Y", col = "red")
grid (col = "blue")

#3.Reemplazar los coeficientes obtenidos de acuerdo al modelo ajustado (En
este caso Thomas y Parresol) y encontrar la variable respuesta (y)

summary (modtTP)

coe <- coef (modtTP)

b1 <- coe [1]

b1

b2 <- coe [2]

b2

b3 <- coe [3]

b3

y <- ((b1*XTZ-1) + b2*sin(2*pi*XTZ) + b3*(1/tan(pi/2*XTZ)))

print(y)

win.graph ()

tp = plot (XTZ ~ y, data = tabla2, type = "o", lwd = 2, main = "Perfil de
Ahusamiento de Thomas y Parresol", xlab = "XTZ", ylab = "Y", col = "black")
grid (col = "gray")
```

```
#-----  
-----
```

```
#Fin del script
```

```
#-----  
-----
```

2. A continuación, se presenta el código generado en Python para la obtención del factor de forma como método de validación entre la relación del volumen real y el volumen aparente (cilindro).

```
"""El siguiente algoritmo calcula el factor de forma de las especies en base  
a la relación entre el volumen real y volumen aparente
```

```
Created by: Bryan Parra and Alonso Ortiz
```

```
Forest Engineering; Industrial University of Santander
```

```
Date: Monday 20-08-2019
```

```
Hour: 15:51 """
```

```
#INICIO
```

```
#Creación de Una Ventana para verificación de Datos
```

```
import math
```

```
from tkinter import *
```

```
from tkinter import messagebox
```

```
ventana=Tk()

ventana.geometry ("500x450+700+65")

ventana.title ("Factor de forma")

ventana.configure (bg = "#444")

colorletra = "#FFF"

#Definiendo a la función encargada de "calcular el volumen aparente del
individuo"

def calcula ():

    try:

        dap = entradaDap.get()

        dap1 = (dap/100)

        dapm = (dap1)**(2)

        At = entradaAt.get()

        pi = math.pi

        va = (pi/4)*((dapm)*(At))

        print ("\nEl volumen aparente del individuo es: " + str(round (va,4)),
"metros cubicos")

        lblVa = Label (text = ("El volumen aparente del individuo es: " + str
(round (va,4)), "metros cubicos"), bg = "#444" , fg = "#FFF", font = ("Times
New Roman",14)).place(x=2,y=215)

    except:

        messagebox.showerror ("Error de sintaxis", message = "Argumentos ausentes o
invalidos")

#Definiendo a la función encargada de "obtener el factor de forma del
individuo"
```

```

def getv ():

    try:

        dap = entradaDap.get()

        dap1 = (dap/100)

        dapm = (dap1)**(2)

        At = entradaAt.get()

        pi = math.pi

        va = (pi/4)*((dapm)*(At))

        vr = entradaVr.get()

        Ff = (vr/va)

        print ("\nEl factor de forma del individuo es: " + str(round (Ff,4)))

        lblFf = Label (text = ("El factor de forma del individuo es: " + str (round
(Ff,4))), bg = "#444" , fg = "#FFF", font = ("Times New
Roman",14)).place(x=2,y=363)

        messagebox.showinfo ("Factor de forma", message = "El factor de forma es: "
+ str (round (Ff,4)))

    except:

        messagebox.showerror ("Error de sintaxis", message = "Argumentos ausentes o
invalidos")

#Creación de labels o etiquetas

lblFF = Label (text = "FACTOR DE FORMA", bg = "#444", fg = "#FFF", font =
("Times New Roman",18)).place(x=138, y=4)

lblVap = Label (text = "* Volumen aparente", bg = "#444", fg = "#FFF", font =
("Times New Roman",16)).place(x=4, y=45)

lblDap = Label (text = "Dap (cm): ", bg = "#444", fg = "#FFF", font = ("Times
New Roman",16)).place(x=4,y=85)

```

```
lblAt = Label (text = "Altura total (m): ", bg = "#444", fg = "#FFF", font =
("Times New Roman",16)).place(x=4,y=125)

lblVr = Label (text = "* Volumen real: ", bg = "#444", fg = "#FFF", font =
("Times New Roman",16)).place(x=4,y=265)

#Creando un campo numérico para Diámetro a la altura del pecho

entradaDap = DoubleVar()

Dap = Entry (ventana, textvariable = entradaDap, font = ("Comic Sans MS",12),
width = 8).place(x=155, y=86)

#Creando un campo numérico para Altura total del individuo

entradaAt = DoubleVar()

At = Entry (ventana, textvariable = entradaAt, font = ("Comic Sans MS",12),
width = 8).place(x=155, y=126)

#Creando un campo numérico para Volumen real (obtenido por integrales en R)

entradaVr = DoubleVar()

Vr = Entry (ventana, textvariable = entradaVr, font = ("Comic Sans MS",12),
width = 8).place(x=155, y=265)

# Creación de Botones

btnCalcular = Button (ventana, text="Calcular",command = calcula, bg =
"#444", fg = "pink", font = ("Times New Roman", 16), relief="raised",
borderwidth=3, width = 6).place(x=270, y=165)
```

```
btnGv = Button (ventana, text="GetFF", command = getv, bg = "#444", fg =
"white", font = ("Times New Roman", 16), relief="raised", borderwidth=3,
width = 6).place(x=270, y=305)
```

```
btnSalir = Button (ventana, text="Salir",command = ventana.quit, fg = "red",
font = ("Times New Roman", 16), relief="ridge", borderwidth="1.5p", width =
5).place(x=429, y=409)
```

```
ventana.mainloop()
```

3. Script generado en Python para generar código QR con información sobre el estado de una plantación o rodal.

```
"""El siguiente algoritmo genera un código QR con el fin de etiquetar la base
de datos forestal como complemento al Proyecto de Grado Forest Engineering;
Industrial University of Santander
```

```
Date: Thursday 21-01-2020
```

```
Hour: 17:46 """
```

```
#Inicio
```

```
#Script ejecutado
```

```
# Importamos el modulo necesario para trabajar con códigos QR
```

```
import qrcode
```

```
# Creamos un objeto codigo QR
```

```
qr = qrcode.QRCode (
    version = 1,
    error_correction = qrcode.constants.ERROR_CORRECT_H,
    box_size = 4,
    border = 4 )
```

```
# Podemos almacenar la información que queremos
```

```
# En el código de manera separada, CREAR LA LISTA CON LA INFORMACION
NECESARIA

info = ("Nombre científico: Eucalyptus globulus"

        "\nNombre comun: Eucalipto"

        "\nFamilia: Myrtaceae"

        "\nEspecie: Exotica"

        "\nCoordenadas: N & E"

        "\nRodal No: MH7"

        "\nEstado fitosanitario: Bueno"

        "\nManejo: Cosecha"

        "\nUsos: Maderable, medicinal, biomasa"

        "\nCubicado: Taper Model of Kozak (1969)"

        "\nVolumen estimado: 150 metros cubicos"

        "\nFactor de forma medio: 0.7")

# Agregamos la información

qr.add_data(info) qr.make(fit=True)

# Creamos una imagen para el objeto código QR

imagen = qr.make_image()

# Guardemos la imagen con la extensión que deseamos

imagen.save('QRTesis.png')
```

5. Recomendaciones

Se recomienda dar continuidad al algoritmo con el fin de lograr mejoras que aumenten la selección de procesos, y así, compilar el mayor número de modelos posibles que también ajuste a especies nativas ampliando el espectro y la calidad del código.

Se recomienda a la Universidad Industrial de Santander Sede Málaga seguir estimulando el uso de software libre enfocado en el desarrollo de las asignaturas y en la creación de los futuros proyectos en las diferentes carreras de nivel superior, garantizando que su uso no incida en aspectos económicos y problemas legales por derechos de autor.

Se recomienda una mayor profundización en el tema de ciencia de datos por parte de la comunidad universitaria, con el fin de alcanzar el ritmo de la competencia impuesta por la educación superior global.

6. Conclusiones

Las líneas de código ejecutadas en el lenguaje de programación R-Project permiten comparar los modelos de ahusamiento empleados, eligiendo como tal al modelo que presente la convergencia de todos los parámetros, bondad de ajuste, mejor comportamiento gráfico fustal y una representación visual de los supuestos de regresión.

Se programó una interfaz gráfica en donde se puede calcular de manera sencilla el factor de forma con una interfaz amigable para el usuario y la creación del código QR en donde esta compilada la información necesaria de las características del individuo o rodal, dando cumplimiento al segundo objetivo específico establecido en el proyecto.

La habilidad de programar debe ser entendida como una necesidad del mundo profesional contemporáneo, siendo esta una herramienta trascendental en el proceso de optimización de resultados de alta precisión en tiempos relativamente pequeños que inciden en el análisis de datos respectivamente.

El uso de software libre permite observar que el tipo de evaluaciones se vuelven más transparentes, mejorando así la comprensión y la aprobación de los resultados obtenidos por las partes involucradas, de esta manera se logra disminuir el sesgo y acceso de la información procesada.

Referencias Bibliográficas

- Aparicio, J., Martínez, M., & Morales, J. (2004). *Modelos lineales aplicados en R*. Elche, España: Centro de Investigación Operativa- Universidad Miguel Hernandez.
- Betancur Calle, J. R., & Pérez García, F. (2013). *Complemento de Excel 2010® para ingeniería económica*. (Ph.D.), Universidad EIA, Envigado, Antioquia.
- Birchler, T. a., Royo, A., & Pardos, M. (1998). La planta ideal: revisión del concepto, parámetros definitorios e implementación práctica. *Forest Systems*, *V(7)*, 109-121.
- Briones Alexander, d. l., & Servando, V. C. (2012). *Convergencia*. Chiapas-Mexico: Instituto Tecnológico de Tuxtla Gutiérrez. Recuperado de <https://sites.google.com/site/metanumericos/home/unidad-1/1-3-convergencia>.
- Bustamante, F. J. (2014). *Estimaciones volumétricas de pinus patula en plantaciones de rápido crecimiento*. (Ph.D.), Instituto Tecnológico del Valle de Oaxaca, Oaxaca.
- Byrne, J. C., & Reed, D. D. (1986). Complex compatible taper and volume estimation systems for red and loblolly pine. *Forest Science*, *V(32)*, 423-443.
- Cancino Cancino, J. O. (2012). *Dendrometría básica*. Chile: Universidad de Concepción. Facultad de Ciencias Forestales.
- Cancino, J. (2006). *Dendrometría Básica*. Santiago de Chile: Concepción: Universidad de Concepción. Facultad de Ciencias Forestales. Departamento manejo de Bosques y Medio Ambiente.
- Carita, G. (2018). *¿Por qué los estudios hídricos o ambientales tienen que usar software libre?* Santiago de Surco - Lima: gidahatari. Recuperado de <http://gidahatari.com/ih-es/por-que-los-estudios-hidricos-ambientales-tienen-que-usar-software-libre>
- Cauas, D. (2015). Definición de las variables, enfoque y tipo de investigación. *Bogotá: biblioteca electrónica de la universidad Nacional de Colombia*, *V(2)*.
- Clutter, J. L., Fortson, J. C., Pienaar, L. V., Brister, G. H., & Bailey, R. L. (1983). *Timber management: a quantitative approach*. USA: John Wiley & Sons, Inc.
- Cornejo Zuñiga, O., & Rebolledo Vega, R. (2016). Estimación de parámetros en modelos no lineales: algoritmos y aplicaciones. *Revista EIA*, *V(13)*, 81-98.
- Coronel de Renolfi, M. (2004). *La programación lineal aplicada al manejo forestal* (1ra ed.). Santiago del Estero, Argentina: UNSE- FCF.

- Correa, N. (2002). *Gráficos estadísticos con R*. Medellín:Universidad Nacional. Recuperado de <http://cran.r-project.org/doc/contrib/grafi3.pdf>
- de-Miguel, S., Mehtätalo, L., Shater, Z., Kraid, B., & Pukkala, T. (2012). Evaluating marginal and conditional predictions of taper models in the absence of calibration data. *Canadian Journal of Forest Research*, *V(42)*, 1383-1394.
- Díaz Muñoz, G. M. (2005). Linear programming as a tool for decision making (programación lineal como herramienta para toma de decisiones financieras). *Universidad de la Rioja*, *V(1)*, 1-8.
- Diéguez-Aranda, U. A. (2012). *Prácticas de Ampliación de Dasometría con R*. Lugo, España: Universidad de Santiago de Compostela, Departamento de Ingeniería Agroforestal.
- Domínguez-Viveros, J., Rodríguez-Almeida, F. A., Núñez-Domínguez, R., Ramírez-Valverde, J. Á., & Ruiz-Flores, A. (2013). Ajuste de modelos no lineales y estimación de parámetros de crecimiento en bovinos tropicarne. *Agrociencia*, *V(47)*, 25-34.
- Duque, R. G. (2011). *Python para todos*. España: Creative Commons Reconocimiento 2.5.
- González, A. V. (2013). *Gráficos estadísticos y mapas con R*. Vigo, España: Ediciones Díaz de Santos.
- González, L. (2018). *Evaluando el error en los modelos de regresión*. USA: LidgiGonzález. Recuperado de <http://lidgigonzalez.com/evaluando-el-error-en-los-modelos-de-regresion/>
- Gujarati, D. N. (1981). *Econometría*. (Tech. rep). Bogotá: McGraw-Hill.
- Halvorsen, H.-P. (2006). *Introduction to Visual Studio and C#*. Norway : University College of Southeast Norway.
- INESEM, R. D. (2004). *Python vs R para el análisis de datos*. Granada: Revista Digital. Recuperado de <https://revistadigital.inesem.es/informatica-y-tics/python-r-analisis-datos/>
- Jáuregui, A. L., & Oviden, P. E. (2004). Estimación bootstrap para el coeficiente de determinación: un estudio de simulación. *REMA*, *V(9)*, 1-14.
- Laporte, A., Reulier, B., Ternoir, S., Chapuis, G., & Kassel, R. (2011). *What is a qrcode*. Francia: Unitag. Recuperado de <https://www.unitag.io/es/qrcode/what-is-a-qrcode>
- Limeres, M. C. (2012). *Regresión lineal simple*. España: Universidad de Santiago de Compostela. Recuperado de http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP-DPTO/MATERIALES/Mat_50140116_Regr_%20simple_2011_12.pdf

- López, A. A., García, J. R., & Navarro, I. I. (2016). Interfaz gráfica en Matlab para el cálculo de criterios de bondad de ajuste. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, V3(5), 13-21.
- López, B. S. (2016). *Regresión Lineal - Ingeniería Industrial*. España:Creativecommos. Recuperado de <https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/pron%C3%B3stico-de-ventas/regresi%C3%B3n-lineal/>
- Mayo, I. C., & Álvarez, R. B. (2011). El profesorado universitario y las tecnologías de la información y de la comunicación (TIC): disponibilidad y formación. *Educatio Siglo XXI*, V(29), 263-302.
- Mi. (2019). *Mínimos cuadrados*. Tutorias Online LLC. Recuperado de <https://miprofe.com/minimos-cuadrados/>
- Minitab, L. L. (2020). *Understanding nonlinear regression*. USA:Minitab. Recuperado de <https://support.minitab.com/en-us/minitab/19/help-and-how-to/modeling-statistics/regression/supporting-topics/nonlinear-regression/understanding-nonlinear-regression/>
- Mora, W. (2016). Como utilizar R en métodos numéricos. *Revista Digital Matemática*, V(16), 1-72.
- Návar-Cháidez, J. d., & Domínguez-Calleros, P. A. (2013). Modelo de incremento y rendimiento: ejemplos y aplicaciones para bosques templados mexicanos. *Revista mexicana de ciencias forestales*, V(4), 8-27.
- Omonte, M., & Valenzuela, L. (2011). Variación radial y longitudinal de la densidad básica en árboles de *Eucalyptus regnans* de 16 años. *Maderas. Ciencia y tecnología*, V(13), 211-224.
- Pacheco, J. M. (2019). *Real Academia Espanola*. Madrid: Real Academia Española. Recuperado de <https://www.rae.es/>
- Paladino, M. (2017). *Introducción al análisis de datos categóricos con R*. Mexico:Instituto Mora. Recuperado de https://www.institutomora.edu.mx/testU/SitePages/martinpaladino/Datos_categoricos.html#datos-categoricos
- Prodan, M. (1997). *Mensura forestal*. Agroamerica. San José, Costa Rica:IICA y GTZ
- Quora. (2014). *What-is-the-CRAN-mirror*. Quora. Recuperado de <https://www.quora.com/What-is-the-CRAN-mirror>
- Raffino, M. E. (2019). *Concepto de algoritmo en informática*.Concepto.de. Recuperado de <https://concepto.de/algoritmo-en-informatica/>

- Ramirez-Alan, O. (2017). *Correlación y Regresión Lineal*. RPubs. Recuperado de <https://rpubs.com/osoramirez/316691>
- Riaño Melo, O. (2017). *Estimación de volúmenes de madera en pie de la especie Pinus patula mediante polinomio único de ahusamiento. Caso de estudio: Finca el Guásimo*. (tesis doctoral), Universidad Distrital Francisco José de Caldas, Bogotá.
- Rodríguez Toro, A., Rubilar Pons, R., Muñoz Sáez, F., Cártes Rodríguez, E., Acuña Carmona, E., & Cancino Cancino, J. (2016). Modelo de ahusamiento para Eucalyptus nitens, en suelos de cenizas volcánicas de la región de La Araucanía (Chile). *Revista de la Facultad de Ciencias Agrarias. Universidad Nacional de Cuyo*, V(48), 101-114.
- Rogel-Salazar, J. (2018). *Data science and analytics with Python*. Reino Unido:Chapman and Hall/CRC.
- Rojas, S., Fernández, H., & Ruiz, J. C. (2016). *Aprendiendo a programar en python con mi computador.*: Caracas, Venezuela: ResearchGate.
- Rossum, G. v. (2017). *El tutorial de Python*. Fred L. Drake Jr. Recuperado de <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>
- Rossum, V., Drake Jr, G., & Fred, L. (2009). *Tutorial Python*.Argentina: Python Software Foundation.
- Salas, C. (2008). ¿Por qué comprar un programa estadístico si existe R? *Ecología austral*, V(18), 223-231.
- Sánchez Alberca, A. (2014). *Bioestadística Aplicada con R y RKTeaching*. España: Creative Commons.
- Santana Sepúlveda, S., & Mateos Farfán, E. (2014). *El arte de programar en R: un lenguaje para la estadística*. México: Instituto Mexicano de Tecnología del Agua. UNESCO. CONAMEXPHI.
- Shipman, J. W. (2013). *Tkinter 8.4 reference: a GUI for Python*. Mexico:New Mexico Tech Computer Center.
- SoloLearn. (2019). *Curso de Python*. San Francisco:Sololearn. Recuperado de <https://www.sololearn.com/Play/Python>
- UNESCO. (1998). *Declaración mundial sobre la educación superior en el siglo XXI: visión y acción*.Paris,Francia:UNESCO. Recuperado de https://unesdoc.unesco.org/ark:/48223/pf0000113878_spa
- Varó, A. M., Sevilla, P. G., & Luengo, I. G. (2014). *Introducción a la programación con Python 3*. España:Universitat Jaume I. Servei de Comunicació i Publicacions.

- Vásquez-Bautista, N., Zamudio-Sánchez, F. J., Alvarado-Segura, A. A., & Romo-Lozano, J. L. (2016). Modelos biométricos forestales en Hidalgo, México: estado del arte. *Revista Chapingo serie ciencias forestales y del ambiente*, V(22), 351-367.
- Vera Correa, M. G. (2017). *Pruebas de bondad de ajuste*. Hidalgo, México: Universidad Autónoma del Estado de Hidalgo. Recuperado de https://www.uaeh.edu.mx/docencia/P_Presentaciones/Sahagun/industrial/2017/Pruebas_de_bondad_de_ajuste.pdf

Apéndices

Apéndice A. Manual del Programador

Introducción

En este anexo se describe la documentación técnica de programación, incluyendo los procedimientos previos para la configuración y el funcionamiento del script.

Entornos de desarrollo

Como se mencionaba anteriormente es necesario tener instalados los entornos de desarrollo integrados como:

- Tinn R, R Studio o Anaconda, entre otros
- Visual Studio Code, Spyder o Geany, entre otros.

Configurando al equipo

1. Iremos al panel de control.
2. Seleccionar la opción de región.

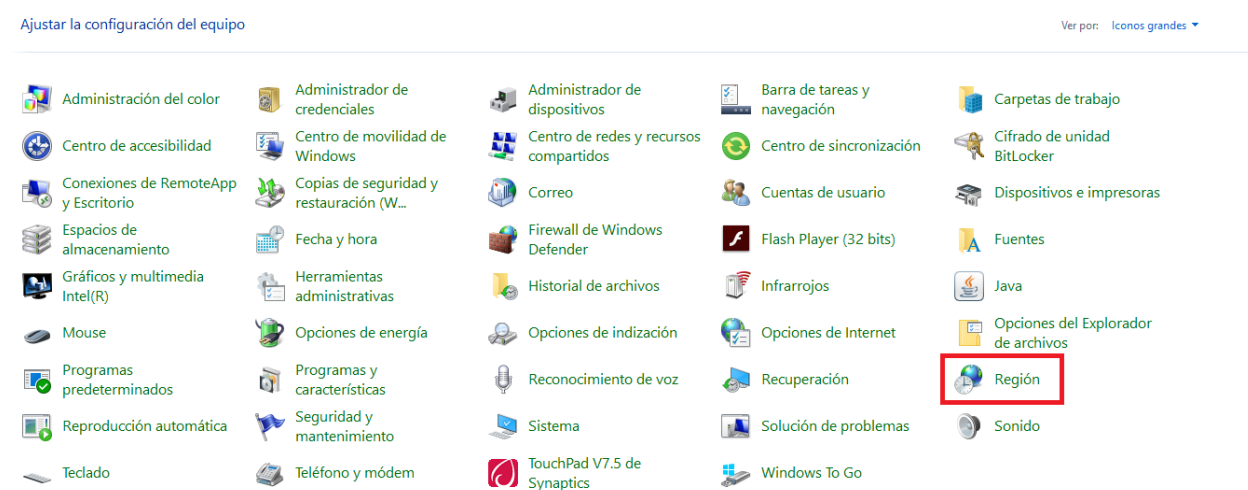


Figura 18. Configuración del equipo.

3. Seleccionar la opción de «configuración adicional» y establecer las configuraciones tal cual como se observa en la figura.

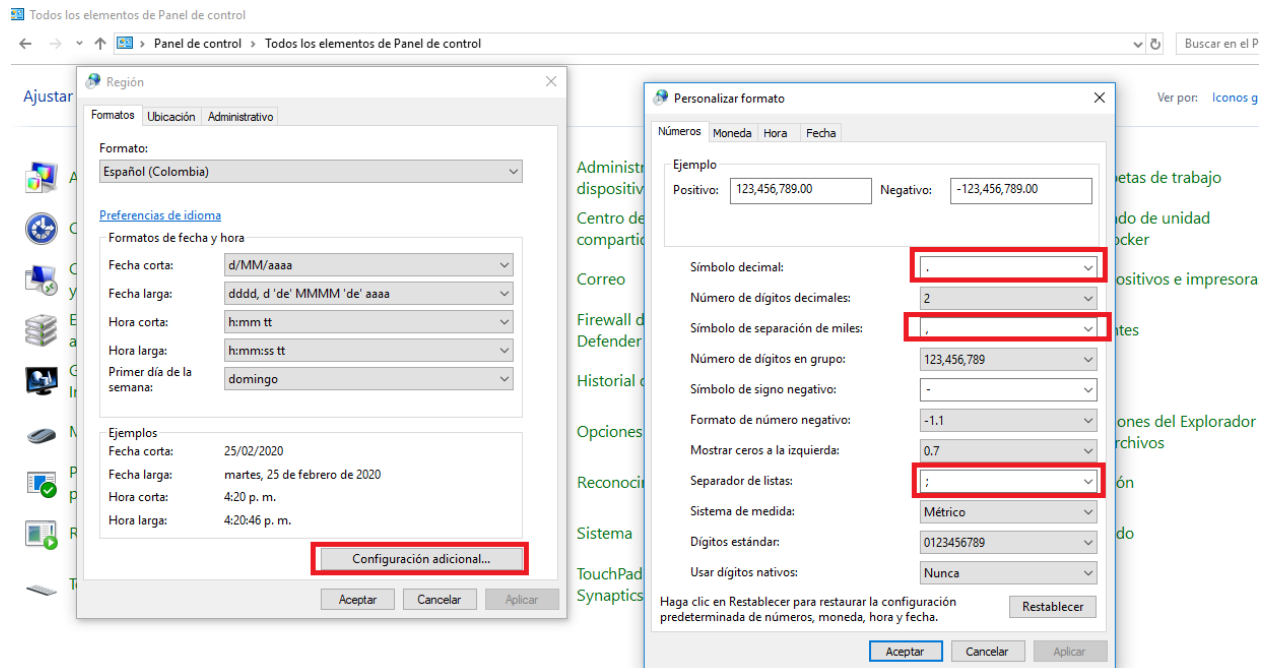


Figura 19. Configuración del equipo.

Esto es necesario ya que las bases de datos están almacenadas en hojas de cálculo y se hace uso de la librería «readxl» de R-Project.

Interfaz y funcionamiento del script

A continuación se describe el funcionamiento del script, la primera parte está programada en R-Project y la segunda en Python, se describe el script haciendo uso del Notebook «Jupyter» de Anaconda, la librería «readxl» permite llamar bases de datos desde hojas de cálculo y la librería «minpack.lm» permite solucionar problemas de mínimos cuadrados no lineales empleando el algoritmo de Levenbert Marquardt.

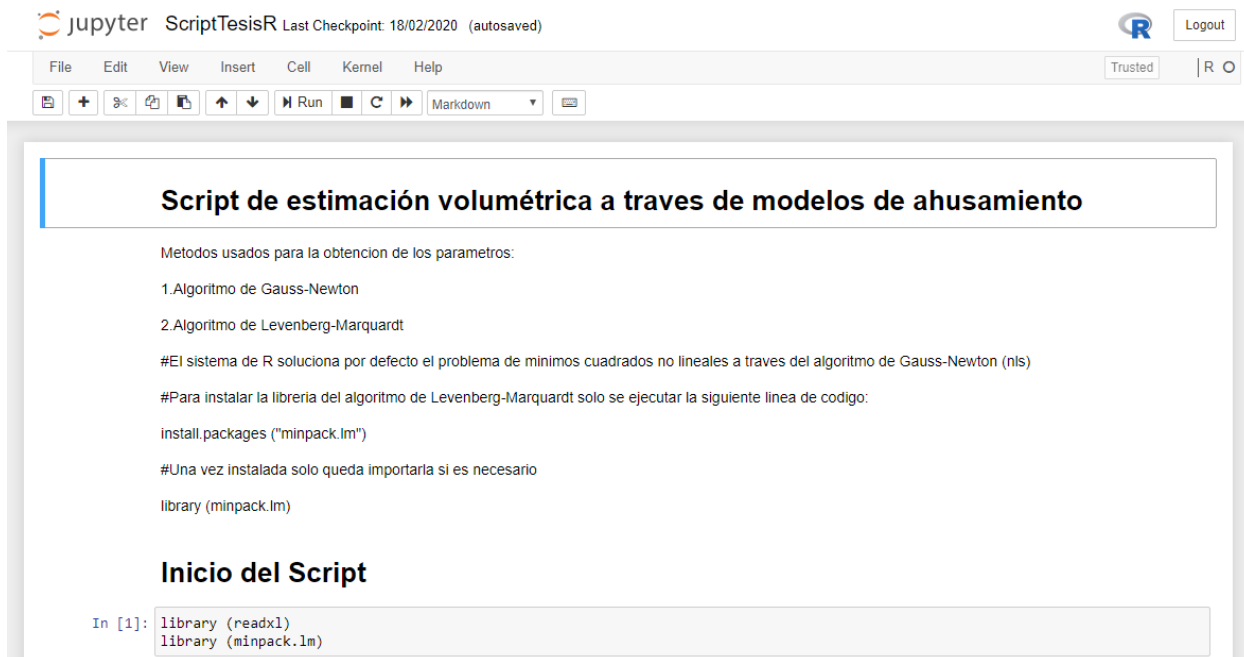


Figura 20. Notebook Jupyter R-Project.

A través del comando «scipen» se puede apagar la notación científica con el fin de observar los resultados obtenidos en notación decimal directamente en R, estas líneas o series de comando son corridas usando el botón «Run».

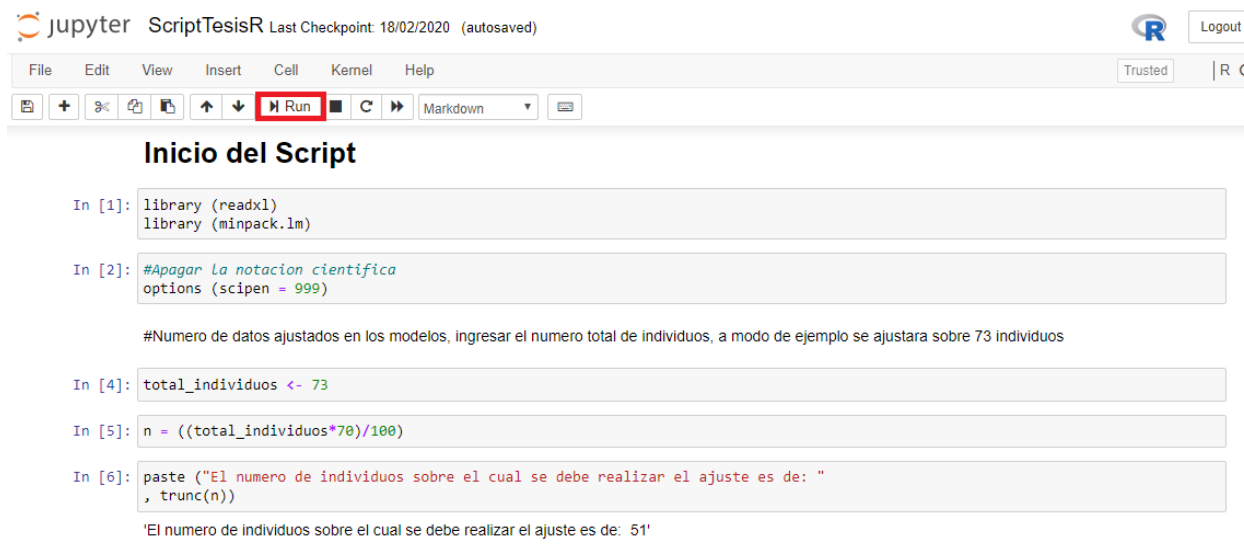
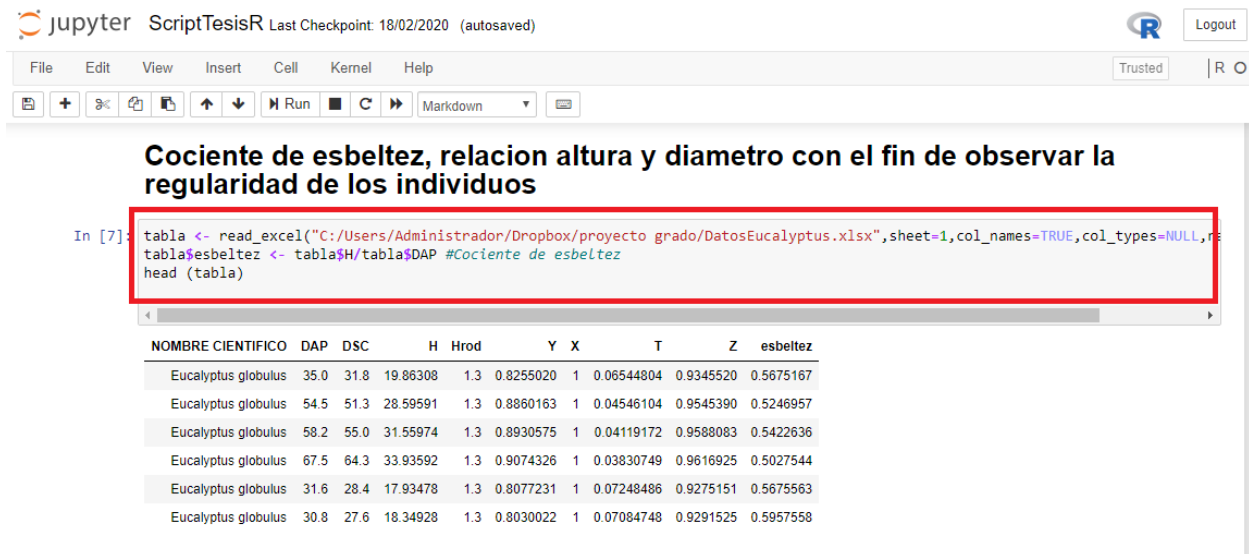


Figura 21. Corriendo las líneas en R.

Mediante el uso de la librería «readxl» y su función «read_excel» como primer argumento debe ir la raíz de la ubicación del archivo con su respectivo formato, en este caso .xlsx, seguido del número de la hoja «sheet» en donde está contenida la información tal cual como es descrito en la figura 22.



The screenshot shows a Jupyter Notebook titled "ScriptTesisR" with a last checkpoint of "18/02/2020 (autosaved)". The code cell contains the following R code:

```
In [7]: tabla <- read_excel("C:/Users/Administrador/Dropbox/proyecto grado/DatosEucalyptus.xlsx", sheet=1, col_names=TRUE, col_types=NULL, read_only=TRUE)
tabla$esbeltez <- tabla$H/tabla$DAP #Cociente de esbeltez
head (tabla)
```

The output of the code is a table with the following columns: NOMBRE CIENTIFICO, DAP, DSC, H, Hrod, Y, X, T, Z, and esbeltez. The data is as follows:

NOMBRE CIENTIFICO	DAP	DSC	H	Hrod	Y	X	T	Z	esbeltez
Eucalyptus globulus	35.0	31.8	19.86308	1.3	0.8255020	1	0.06544804	0.9345520	0.5675167
Eucalyptus globulus	54.5	51.3	28.59591	1.3	0.8860163	1	0.04546104	0.9545390	0.5246957
Eucalyptus globulus	58.2	55.0	31.55974	1.3	0.8930575	1	0.04119172	0.9588083	0.5422636
Eucalyptus globulus	67.5	64.3	33.93592	1.3	0.9074326	1	0.03830749	0.9616925	0.5027544
Eucalyptus globulus	31.6	28.4	17.93478	1.3	0.8077231	1	0.07248486	0.9275151	0.5675563
Eucalyptus globulus	30.8	27.6	18.34928	1.3	0.8030022	1	0.07084748	0.9291525	0.5957558

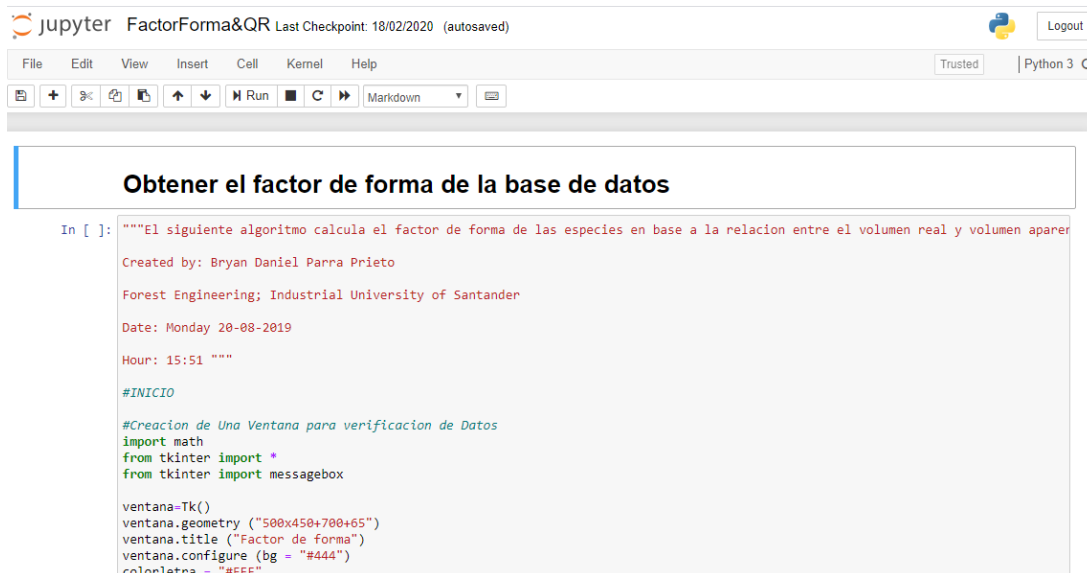
Figura 22. Subida de la base de datos.

A continuación se hace una breve descripción de los argumentos y su respectivo funcionamiento:

- **\$**: este símbolo permite extraer información referente ya sea de una variable generada, calculada o almacenada en la base de datos.
- **head**: permite hacer una visualización en consola de la tabla o dataframe generado.
- **data**: permite referenciar los datos de donde se está obteniendo la información.
- **nls y nlsLM**: estas funciones permiten solucionar problemas de mínimos cuadrados mediante el algoritmo de Gauss Newton y Levenbert Marquardt.
- **summary**: permite observar las características estadísticas calculadas de las variables programadas.

- *integrate*: soluciona la integral definida o programada, obteniendo como primer argumento la función, seguido del límite inferior y terminando con el límite superior de la integral.
- *residuals*: permite obtener los residuales calculados.
- *win.graph ()*: permite ir almacenando las gráficas que han sido generadas anteriormente sin que se pierdan.
- *lm*: permite efectuar una regresión lineal entre una o más variables dependientes e independiente.
- *plot*: esta función genera gráficos lineales sencillos.
- *abline*: traza una línea de tendencia.
- *AIC Y BIC*: permite calcular el índice de criterio de información de Akaike y el índice de criterio de información Bayesiano.
- *grid*: genera una grilla en los gráficos.

Como segunda parte se hace uso del lenguaje de programación Python, se describe el script haciendo uso del Notebook «Jupyter» de Anaconda.



```
In [ ]: """El siguiente algoritmo calcula el factor de forma de las especies en base a la relacion entre el volumen real y volumen aparente
Created by: Bryan Daniel Parra Prieto
Forest Engineering; Industrial University of Santander
Date: Monday 20-08-2019
Hour: 15:51 """

#INICIO

#Creacion de Una Ventana para verificacion de Datos
import math
from tkinter import *
from tkinter import messagebox


ventana=Tk()
ventana.geometry ("500x450+700+65")
ventana.title ("Factor de forma")
ventana.configure (bg = "#444")
colorletra = "#FFF"
```

Figura 23. Notebook «Jupyter Python».

A continuación se hace una breve descripción de los módulos empleados y su respectivo funcionamiento:

- *math*: uno de los tantos módulos matemáticos especializados en Python.
- *tkinter*: un módulo en Python que permite generar interfaces gráficas.
- *qrcode*: genera la salida del código QR con la información almacenada en la extensión deseada, en este caso .png

Para la generación del código QR es necesario instalar el modulo que permite generar el código QR, empleando el «pip install qrcode y pip install pillow en la consola o el cmd (símbolo del sistema)», el proceso es descrito en la figura 24.

 Símbolo del sistema

```
Microsoft Windows [Versión 10.0.17134.1246]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Administrador>pip install qrcode
```

Figura 24. Instalación de los módulos en la consola del cmd para generar el código QR.

Apéndice B. Manual del usuario

Requisitos mínimos

Para la ejecución del script, el computador personal debe contar con los siguientes requisitos mínimos para la instalación de los programas encargado de la ejecución del código que son Python y R Project.

Tabla 10.

Requisitos mininos.

Componentes	Requerimientos
Procesador	Intel o Amd mínimo con 2 núcleos de procesamiento
RAM	Mínimo 2 Gb
Disco Duro	20 Gb para la instalación de los programas para ejecutar el código
Sistema operativo	Todas las versiones de Windows, Linux y Mac

Pasos para la ejecución del script creado en R Project.

1. Para poder ejecutar el código creado en R Project hay que tener en cuenta de tener instaladas las librerías en especial (readxl) que es la encargada de cargar la base de datos de Excel en el programa y la librería (minpack.ml) que se encarga de hacer la convergencia mediante el algoritmo de Levenberg Marquardt.
2. En la ejecución del código se utilizó la herramienta Jupyter Notebook de la distribución de Anaconda Python en el sistema operativo de Windows, el cual se puede descargar de la página oficial <https://www.anaconda.com/distribution/>, en donde esta herramienta necesita conexión a internet para que se pueda ejecutar

3. Le damos clic al botón para abrir la interfaz de Jupyter Notebook, que se encuentra ubicada en la carpeta de Anaconda en la barra de los programas instalados de Windows como se muestra en la figura 25.

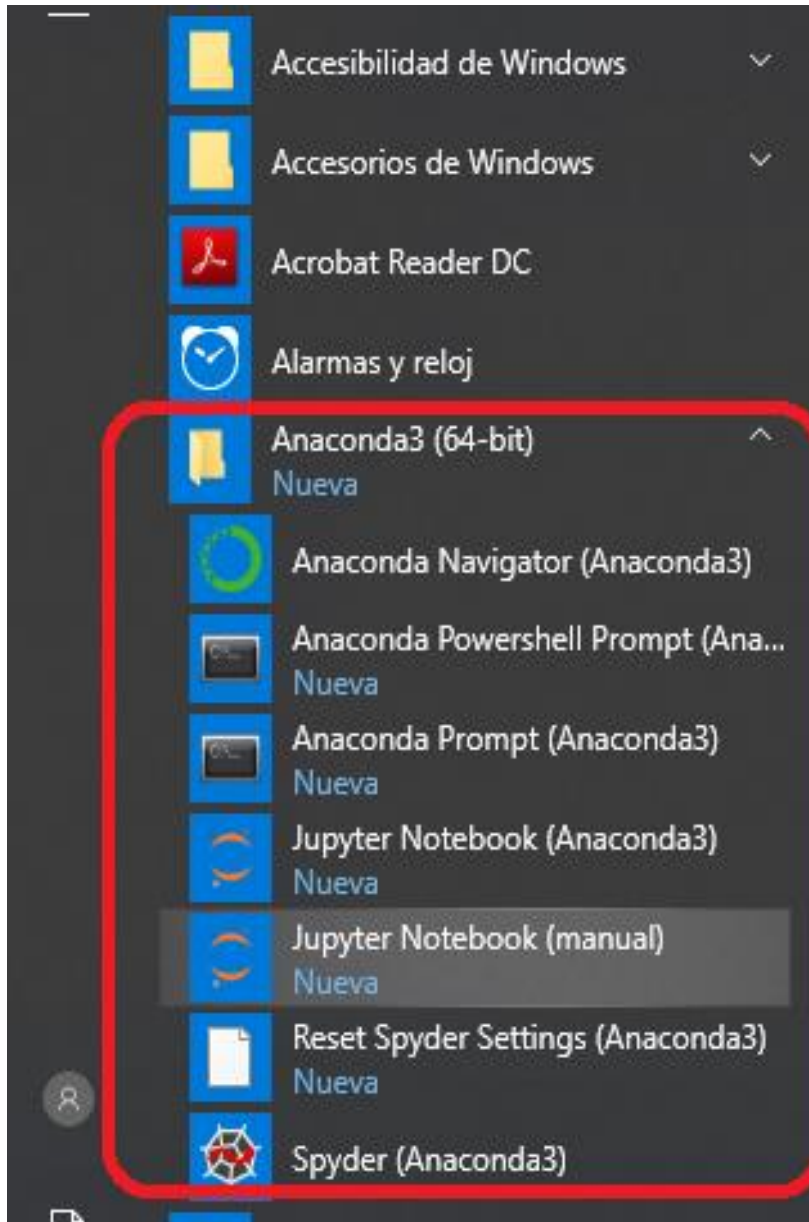


Figura 25. Entorno Jupyter.

4. Una vez abierta la interfaz, procedemos a conocer como se crea un nuevo proyecto de R en el Jupyter en donde se ejecutará la primera parte del script, en donde le damos un clic en “new” y escoge la opción “R”.

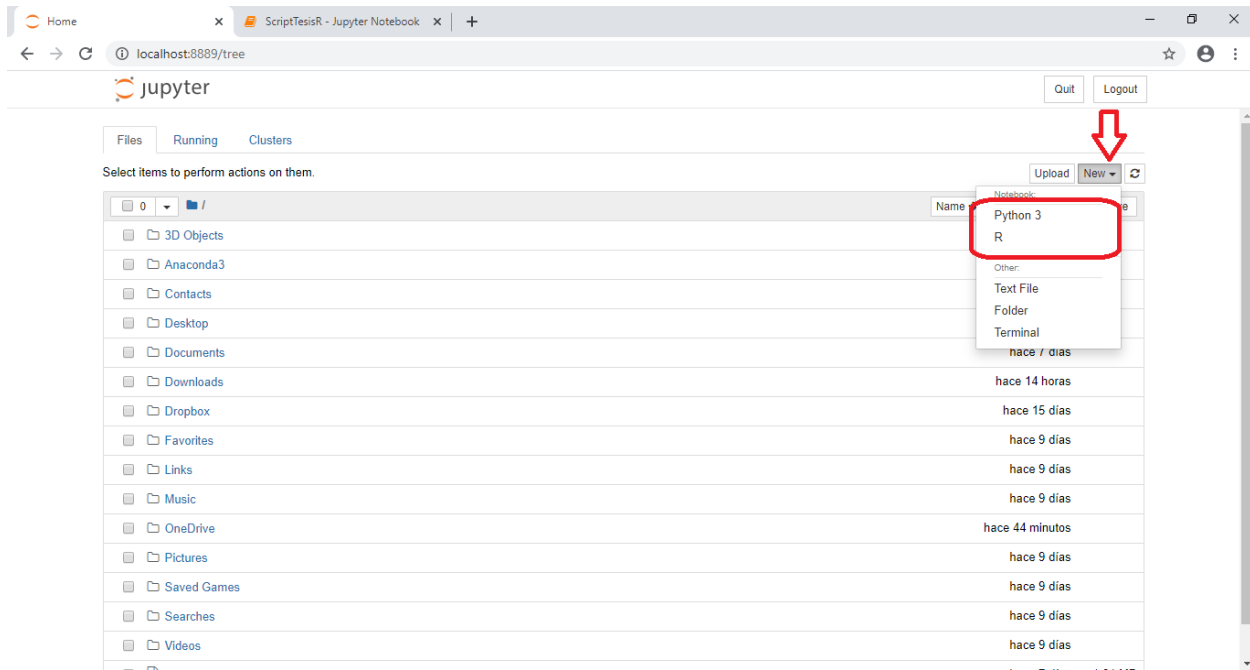


Figura 26. Herramienta Jupyter.

5. Una vez abierto el proyecto, procedemos a escribir el código en la celda que se genera por defecto como se muestra en la figura 27, en donde empezamos escribiendo las librerías para que a la hora de correrlo no se genere ningún error.

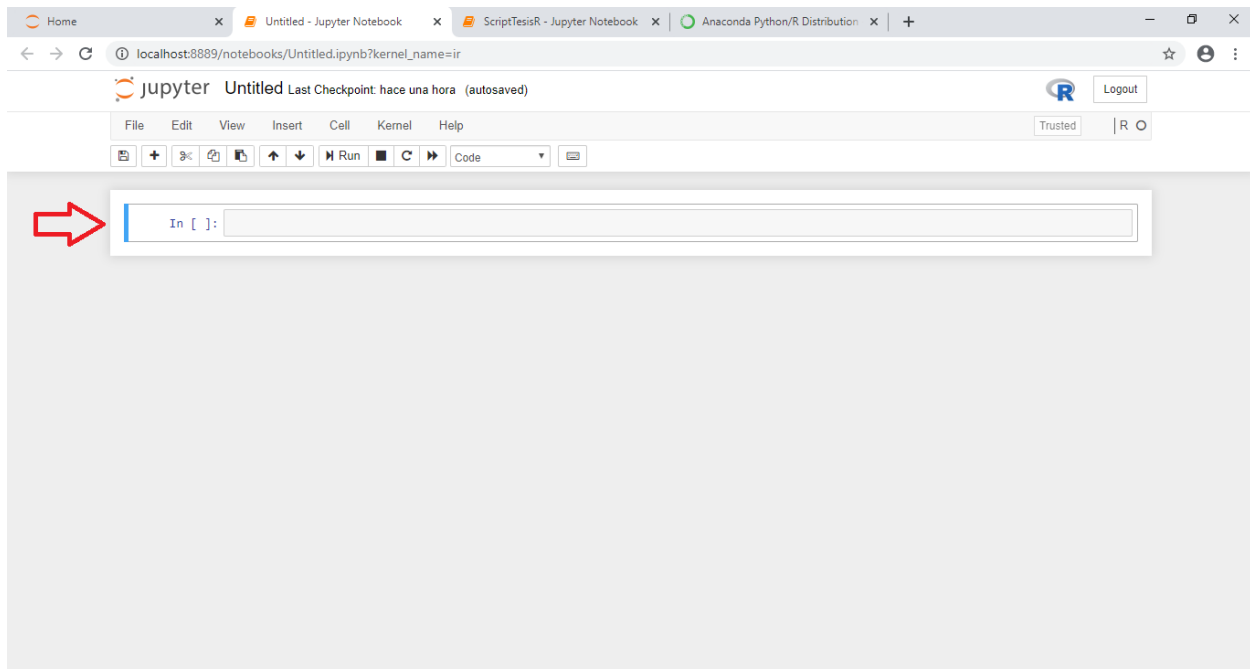


Figura 27. Entorno del primer trabajo.

4. Una vez cargado las librerías, se inicia con la escritura del código respectivo, manteniendo el orden que se estableció en el documento general.

5. Después de escribir el código fuente o script, se da clic en “Run” en donde se iniciara la ejecución del código, tal cual como se observa en la figura 28.

Se debe tener en cuenta las direcciones o rutas de los documentos para evitar errores a la hora de cargar la base de datos, en este caso la base de dato se encuentra almacena en una nube web que es Dropbox.



```
localhost:8888/notebooks/Dropbox/campamento/ScriptTesisR.py?nb#

jupyter ScriptTesisR Last Checkpoint: 18/02/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Help Not Trusted | R O

+ + + + + Run + + + + +

Inicio del Script

In [2]: library (readxl)
library (minpack.lm)

In [3]: #Apagar la notacion cientifica
options (scipen = 999)

#Numero de datos ajustados en los modelos, ingresar el numero total de individuos, a modo de ejemplo se ajustara sobre 73 individuos

In [4]: total_individuos <- 73

In [5]: n = ((total_individuos*70)/100)

In [6]: paste ("El numero de individuos sobre el cual se debe realizar el ajuste es de: "
, trunc(n))

'El numero de individuos sobre el cual se debe realizar el ajuste es de: 51'

Cociente de esbeltez, relacion altura y diametro con el fin de observar la
regularidad de los individuos

In [6]: tabla <- read_excel("C:/Users/Administrador/Dropbox/proyecto grado/DatosEucalyptus.xlsx", sheet=1, col_names=TRUE, col_types=NULL, na
tabla$esbeltez <- tabla$H/tabla$DAP #Cociente de esbeltez

localhost:8888/notebooks/Dropbox/campamento/ScriptTesisR.py?nb#
```

Figura 28. Códigos en Jupyter.

Pasos para la ejecución del script creado en Python

Para la ejecución del script creado en Python se siguieron los mismos pasos en la ejecución del script creado en R Project, los scripts desarrollados son el cálculo del Factor de Forma y la etiqueta QR.

1. Le damos clic al botón para abrir la interfaz de Jupyter Notebook, que se encuentra ubicada en la carpeta de Anaconda en la barra de los programas instalados de Windows como se muestra en la figura 25.
2. Una vez abierta la interfaz, procedemos a conocer como se crea un nuevo proyecto de Python en el Jupyter en donde se ejecutará la primera parte del script, en donde le damos un clic en “new” y escoge la opción “Python” como se muestra en la figura 26.
3. Una vez abierto el proyecto, procedemos a escribir el código en la celda que se genera por defecto como se muestra en la figura 27, en donde empezamos escribir el modulo para que a la hora de correrlo se genere una aplicación más agradable.
4. Una vez cargado el modulo, se inicia con la escritura del código respectivo, manteniendo el orden que se estableció en el documento general.
5. Después de escribir el código fuente o script, se da clic en “Run” en donde se iniciará la ejecución del código.

Se debe tener en cuenta las direcciones o rutas de los documentos o archivos que se exporten para evitar errores a la hora de cargar los datos, en este caso los datos se encuentran almacenados en una nube web, que es Dropbox.