

Universidad Manuela Beltrán

**Diplomado en Bioestadística
Modulo I: Introducción a R**

Elaborado por

William Javier Rodríguez Cruz

**PhD en Ciencias
(Física)**

Bogotá, Colombia
Mayo 2024.

©2024. - William Javier Rodríguez Cruz

Derechos Reservados

Contenido

1	Lenguajes de programación orientados a objetos	1
1.1	Preliminares	1
1.2	Lenguaje de programación R	2
1.2.1	Instalación	2
1.2.2	Posit cloud	2
1.2.3	Creación de un proyecto	3
1.3	Medidas de tendencia central	3
1.3.1	Gráfico de barras	5
1.3.2	Transformación de variables	6
1.3.3	Histogramas	7
1.3.4	Box-Plot (Diagrama de caja)	7
1.4	Actividad de comprobación de trabajo autónomo 2	9
	Bibliografía	10

Capítulo 1

Lenguajes de programación orientados a objetos

1.1 Preliminares

La versatilidad es sinónimo de lenguajes de programación orientados a objetos, este tipo de lenguajes son análogos a una superficie comercial donde es posible conseguir una gran variedad de objetos clasificados en lineales, como por ejemplo: Aseo, básicos, licores, etc. Estos objetos así abandonemos la superficie mantienen sus características y su función. Naturalmente, emergen objetos idénticos etiquetados con marcas diferentes que cumplen la misma función, en esencia, es el mismo objeto. Proyectando esta analogía en el contexto de la programación orientada a objetos, desde el pragmatismo, un programa orientado a objetos corresponde a un software cuyas líneas de código, en su mayoría están reservadas a la invocación de objetos que cumplen una función específica.

La función resulta imprescindible, es la característica que define el objeto que debo adquirir en el súpermercado y conforme se agota basta con volver a comprarlo; en programación el proceso es similar, después de creado el objeto su uso es ilimitado y basta con unas líneas de código para usarlo. Esta simple invocación es la característica primordial que define un lenguaje de programación orientado a objetos, es decir, no se debe crear un objeto de «ceros», esto no significa que esté prohibido. Sin embargo, existe una múltiple variedad de objetos que se puede adaptar a las necesidades del programador.

Existen diferentes tipos de lenguajes de programación orientados a objetos: C++, Java, R, Python, Julia y Matlab, por mencionar algunos. No todos son de acceso libre, lo cual supone un obstáculo respecto a la adquisición del software. La selección del software está condicionada al enfoque o tipo de análisis que requiere el programador. En este curso el enfoque es el análisis estadístico de datos, dicha característica nos lleva a seleccionar dos posibles lenguajes: R o Python. Ambos lenguajes son de

acceso libre (Open Source). Por un lado, R está reservado para análisis estadísticos robustos, cuenta con una incommensurable cantidad de paqueterías y librerías que permiten la simulación y análisis estadístico con formalismos sólidos que lo convierten en la primera opción para economistas, estadísticos y bioestadísticos. Por otra parte, Python es un lenguaje que amplía las fronteras y habilita la opción de simular sistemas estadísticos y de otra naturaleza, como los asociados a las ciencias básicas. Actualmente Python es explotado en trending, Big Data, inteligencia artificial y machine learning.

La selección entre R y Python resulta evidente. La balanza se inclina para R. El álgebra lineal es la piedra angular de los lenguajes de programación orientados a objetos. Todas las operaciones desarrolladas se gobiernan por el álgebra lineal, por lo que se sugiere al estudioso repasar a este respecto, con el fin de que comprenda la forma en que opera R. No es un requisito, pero si garantiza una mejor comprensión en el momento de desarrollar líneas de código.

1.2 Lenguaje de programación R

1.2.1 Instalación

El enlace <https://cloud.r-project.org/> guía el proceso de instalación de R y el entorno RStudio en donde se desarrolla la sintaxis. Para la instalación debe seleccionar su sistema operativo y seguir las indicaciones.

1.2.2 Posit cloud

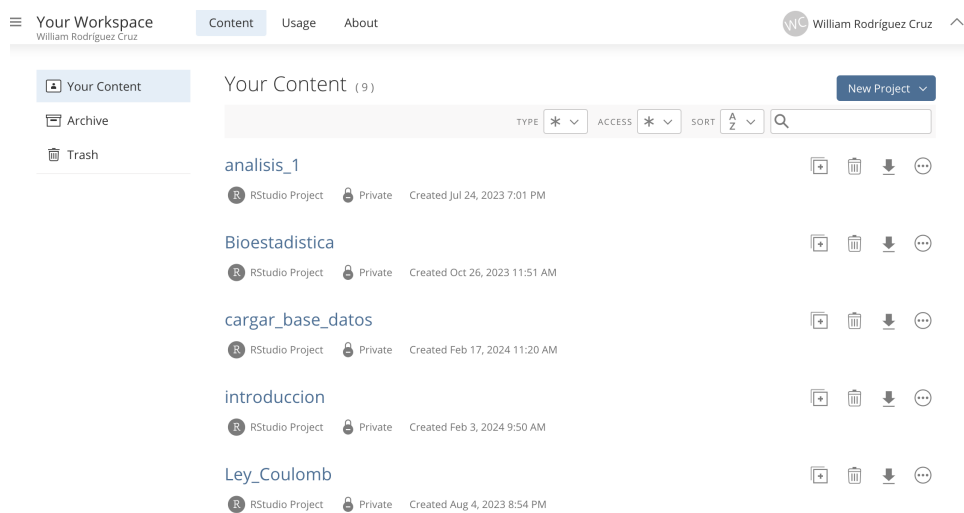


Figura 1.1: Pantalla de inicio plataforma posit cloud.

Una opción alternativa es abrir una cuenta en [Posit Cloud](#) registrarse y acceder a la versión gratuita. Cuando logre ingresar la primera ventana debe ser similar a la mostrada en la fig1.1. ejecute el botón «New Project» y seleccione la segunda «opción». Si se inclinó por la primera opción: instalar el software en su computadora, ahí también podrá crear un proyecto.

1.2.3 Creación de un proyecto

Iniciamos creando un proyecto que se va a titular: «regresion lineal». La creación de un proyecto es análoga a una carpeta y los script corresponden a las hojas que contiene la carpeta. La creación del proyecto los lleva a una ventana como la mostrada en la 1.2. para modificar el nombre del proyecto debe dar clic en «untitled project» y

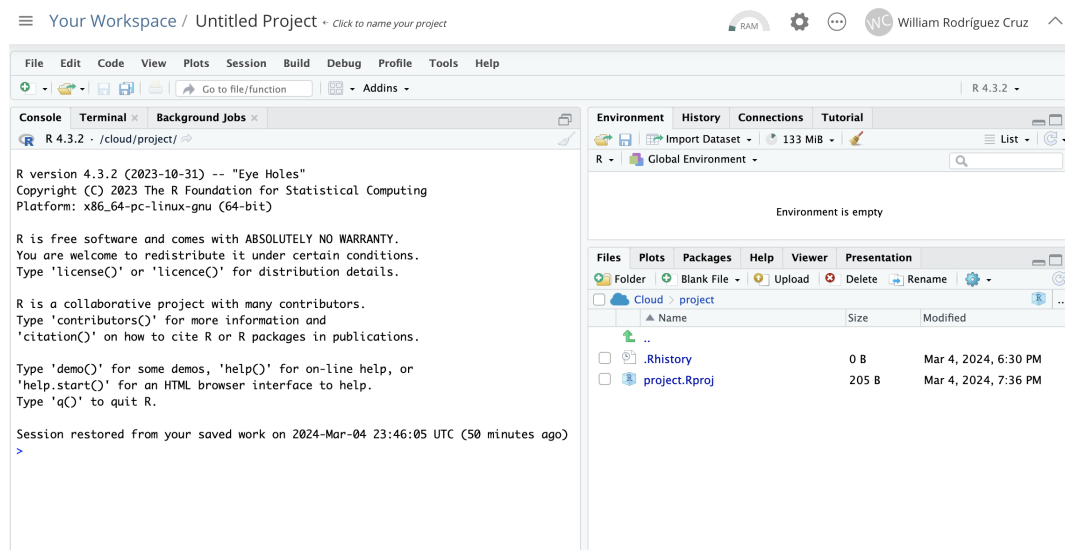


Figura 1.2: Pantalla de inicio de un proyecto en R.

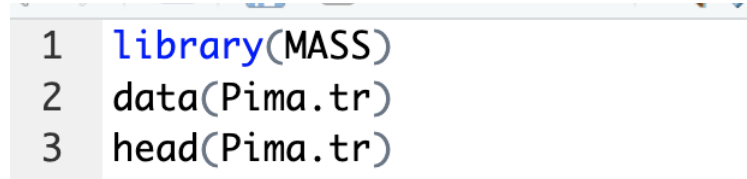
digitar el título del proyecto «regresion lineal». Una vez titulado el proyecto creamos el primer script ejecutando el icono localizado en la parte superior izquierda, justo abajo del botón «file» se desplegará un menú seleccione la primera opción «R script», automáticamente se creará un script sin titulo, de clic sobre el diskette que aparece diagonal abajo y nómbrelo como «tendencia central».

1.3 Medidas de tendencia central

El propósito de esta sección reposa en la simulación de medidas de tendencia central, se parte del supuesto que los estudiosos y las estudiosas reconocen y aplican las medidas de tendencia central: Media, mediana, varianza y desviación estándar.

No obstante, se recomienda el libro de texto [2]. Para el estudio de medidas de tendencia se inicia el estudio de una base de datos real.

Después de la creación de su proyecto y script, escriba las siguientes líneas de código, e intente describir la función de cada una de las tres líneas de código. Después,



```
1 library(MASS)
2 data(Pima.tr)
3 head(Pima.tr)
```

Figura 1.3: Explorando una base de datos

de comprender las tres líneas de código y ejecute el comando «`help(Pima.tr)`» que describe el contenido de la base de datos que se va a estudiar. Antes de iniciar con el análisis de tendencia central, se presentan cuatro comandos útiles para indagar por el tamaño de la base de datos, las variables y de qué naturaleza son estas,

1. `class()`
2. `length()`
3. `dim()`
4. `names()`,

en los parentesis de cada una de las cuatro instrucciones se debe relacionar la base de de datos bajo estudio, en nuestro caso «Pima.tr». Por favor ejecute, las cuatro instrucciones e identifique la función de cada una. Si ha tenido éxitos en identificar la función de cada una de las cuatro instrucciones, podrá responder las siguientes preguntas

1. ¿De qué clase es Pima.tr?
2. ¿Qué variables se están estudiando?
3. ¿Cuántos datos reposan en Pima.tr?
4. ¿Cuántas columnas tiene Pima.tr?

En estadística es esencial comprender la base de datos, es decir, identificar qué variables son cualitativas y cuantitativas. La instrucción «`class(Pima.tr$variable)`» permite identificar la clase de las variables, ejecute el comando `class()` para las ocho variables e identifique la sintaxis y funcionalidad de este comando. Para finalizar esta fase exploratoria ejecute la instrucción «`summary()`» y describa la función de esta instrucción y analice los resultados que arroja dicho comando.

1.3.1 Gráfico de barras

El diagrama icónico en el análisis descriptivo es el diagrama de barras en R se obtiene con la instrucción

- `barplot(frecuencia, xlab = "Tipo", ylab = "frecuencia", main = "Gráfico de barras")`

Si el código se ejecutó con éxito debe obtener la siguiente gráfica, para complementar

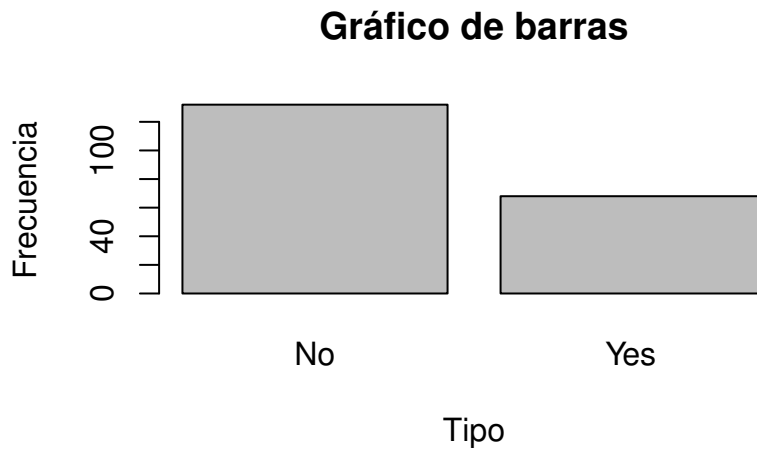


Figura 1.4: Diagrama de barras, obtenido en R para la base datos Pirma.tr.

la información resulta útil reportar la información en términos de la frecuencia relativa que arroja el porcentaje o proporción en cada categoría y se calcula como se sigue,

$$p_c = \frac{n_c}{n}, \quad (1.1)$$

entonces, se requiere cada unas de las frecuencias n_c alojadas en el objeto «frecuencia» y el total de la muestra n . La suma de todas las frecuencias corresponde al total del tamaño de la muestra, es decir,

$$n = \sum_c n_c \quad (1.2)$$

Esta suma en R se calcula por medio de la instrucción

- `n <- sum(frecuencia)`

y debe arrojar un total de 200. El paso natural es calcular la frecuencia relativa, dividiendo cada una de la frecuencias entre el tamaño total de la muestra. En R creamos una tabla de frecuencias, por medio de la siguiente instrucción:

- `frecuenciarelativa <- frecuencia/n`

y con el comando `round()` con un límite de dos cifras decimales, obtenemos la frecuencia relativa cuyo valor máximo al sumar cada frecuencia relativa no debe superar a 1, es decir,

- `round(frecuenciarelativa,2)`

Una opción alternativa consiste en omitir la línea de código con `round()` y sólo imprimir el objeto «frecuenciarelativa». Ahora, multiplicando por 100 a la instrucción `round()`, como se sigue,

- `round(frecuenciarelativa,2)*100`

se obtienen los porcentajes asociados a las frecuencias relativas, que se pueden graficar en un diagrama de barras con la instrucción y obtener la gráfica que se muestra en la fig.1.5

- `barplot(frecuenciarelativa, xlab = "clase", ylab = "Frecuencia relativa", main = "Gráfico de barras (frecuencia relativa)")`

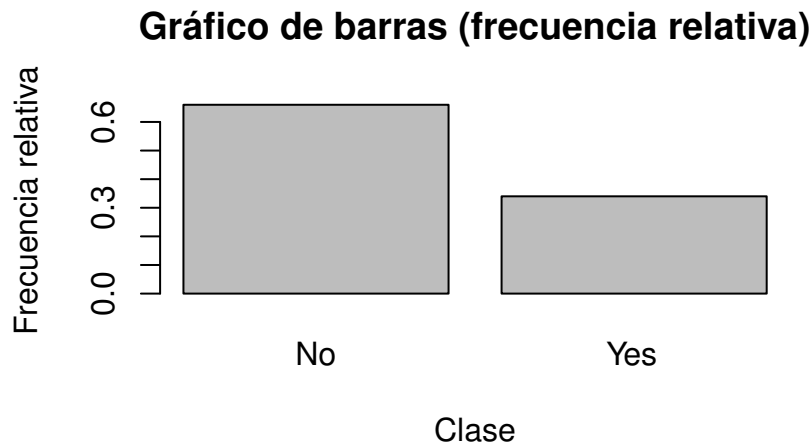


Figura 1.5: Diagrama de barras, obtenido en R para la base datos Pirma.tr, frecuencia relativa.

1.3.2 Transformación de variables

Con el objetivo de transformar variables de tipo numérico a factor, estudiemos la base de datos «birthwt», donde si se indaga por la clase de la variable «smoke» se obtiene que es entera (integer), pero almacena ceros y unos, entonces resulta útil dicotomizarla, es decir, transformarla en una variable categórica de tipo factor. Si

escribimos la instrucción `is.factor(birthwt$smoke)`, previamente cargando la base de datos con la instrucción `data(birthwt)`, se obtiene como resultado `FALSE`, lo que indica que la variables no es categórica, entonces digitando,

- `birtwt$smoke <- factor(birthwt$smoke)`

La variable se transforma en cualitativa, como se verifica con `is.factor(birthwt$smoke)`.

1.3.3 Histogramas

Un segundo gráfico es el histograma y es muy utilizado para detectar la distribución de datos, trabajando nuevamente con `Pima.tr` resulta útil crear un histograma que la frecuencia contenga la edad. La siguiente instrucción permite obtener un histograma,

- `hist(Pima.tr$age, freq = TRUE, xlab = "Edad", ylab = "Frecuencia", col = "green", main = "Histograma de frecuencia para la edad")`,

como el mostrado en la imagen 1.6

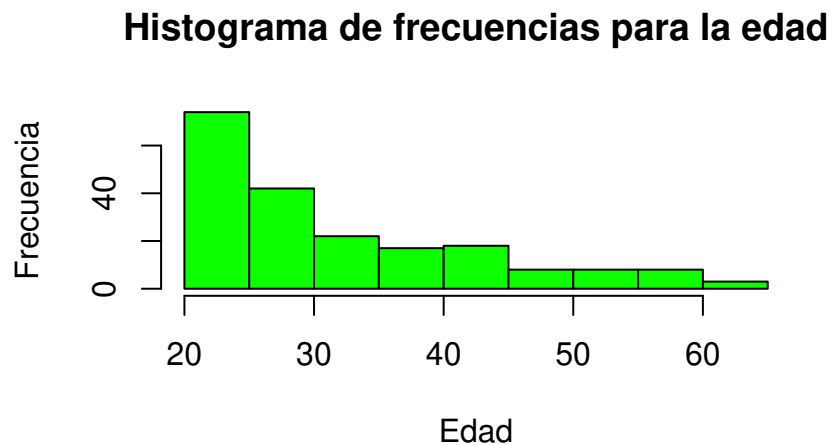


Figura 1.6: Histograma para la edad, obtenido en R para la base datos `Pima.tr`.

1.3.4 Box-Plot (Diagrama de caja)

Este diagrama resulta útil para indagar por la simetría de la distribución de datos, los datos atípicos y la distribución de los cuartiles. Cuando ejecutamos la línea de código «summary» para la base de datos `Pima.tr`, para las variables numéricas aparecieron los cuartiles, valores máximos y mínimos, la mediana y la media, toda esta información resulta mas sencilla de observar en un diagrama Box-Plot. Por ejemplo, para la variable «BMI», se obtiene el Box-Plot digitando

- `boxplot(Pima.tr$bmi, ylab = "BMI")`

por defecto el diagrama sale orientado de forma vertical. No obstante si se quiere orientar de forma horizontal basta con agregar en la última instrucción separado por una coma la instrucción `horizontal = TRUE`. Los resultados obtenidos se muestran en las figuras 1.7 v 1.8. respectivamente.

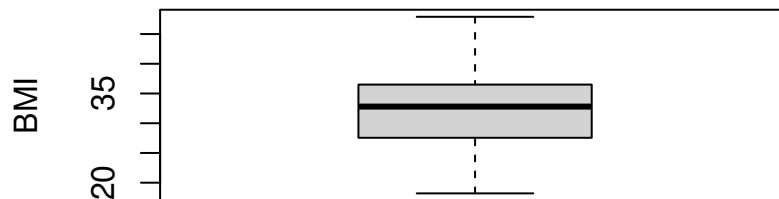


Figura 1.7: Box-Plot, obtenido en R para la base datos Pima.tr.

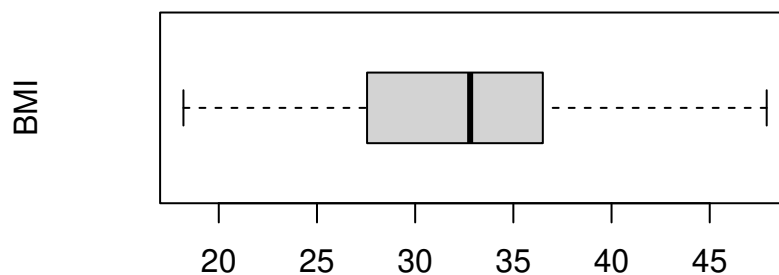


Figura 1.8: Box-Plot, obtenido en R para la base datos Pima.tr, orientación horizontal.

Con este diagrama finalizamos el análisis de tendencia central. Naturalmente, el estudioso puede continuar explorando el lenguaje de programación R, que como se ha demostrado resulta versátil para el análisis estadístico de datos.

1.4 Actividad de comprobación de trabajo autónomo 2

Desarrollar el cuestionario que reposa en el aula canvas en la unidad 1.

Bibliografía

- [1] <https://es.overleaf.com/learn>.
- [2] Shahbaba. B. (2011). Biostatistics with R: An introduction to Statistics Through Biological Data. Springer: Baltimore.