**FASE 2 – Data Science**

**Programación y estadística con R**

**Introducción a R y Software**

**Prework.**

**1.- Introducción.**

* Instalar R
* Instalar R Studio
* Instalar Paquetes en R Studio

Comando de instalación de paquetes en R Studio

> install.packages(“dplyr”)

> library(dplyr)

> ?mutate // Esta línea te sirve para obtener ayuda sobre ese comando.

**2.- ¿R o R Studio?**

* R es un lenguaje de programación
* R Studio es un Entorno Integrado de Desarrollo (IDE), el cual brinda herramientas para poder escribir nuestro código de manera amigable y poder administrar nuestros proyectos.

Imagen que contiene Patrón de fondo

Descripción generada automáticamente

R puede vivir sin R Studio, pero R Studio no puede vivir sin R.

**3.- Empezando con R Studio.**

Al abrirla, podemos ver tres elementos principales:

1. La consola
2. El entorno e historia
3. Files/Plots/Packages/Help/Viewer

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

**4.- Mi primera línea de código.**

Escribe la siguiente línea de código en tu consola:

print('Hola R')

[1] "Hola R"

Imprimió el texto Hola R. Nota: “No es necesario utiliza la función print cuando se requiera imprimir una variable, con su nombre es suficiente”.

Es de resaltar que en R el uso de ‘ ’ es idéntico al de utilizar “ ”.

**5.- R como Calculadora**

Las siguientes son Operaciones básicas y la manera de escribir el código en R.

Tabla

Descripción generada automáticamente

**6.- ¿Cómo escribir números pequeños y grandes?**

Hay que tener en cuenta que cuando escribimos números pequeños o grandes, R cambia los números a la **notación e**. Podemos ver el siguiente ejemplo:

+ 10000000

[1] 1e+07 // == 1x107

+ 0.0000001

[1] 1e-07 // == 1x10-7

En resumen, la notación e nos dice que cualquier número *a* puede ser expresado como *aey*, que representa al número *a x 10y.*

Pero R tiene un límite. Al intentar representar un número mucho más grande o mucho más pequeño, entonces lo hace con infinitos. Por ejemplo:

+ 10000^100

[1] Inf

+ -10000^100

[1] -Inf

+ 1/0

[1] Inf

+ -1/0

[1] -Inf

**7.- Orden de operaciones.**

R, respeta al ejecutar código un orden de expresiones matemáticas. Por lo tanto, hay que tenerlas presentas, este orden lo enlistamos a continuación:

* Paréntesis
* Exponentes
* Multiplicación
* División
* Suma
* Resta

**8.- Help y Stackoverflow**

Para buscar ayuda sobre algúna función o comando podemos hacer uso de lo siguiente:

> ?comando //Ayuda sobre el comando y sus argumentos

> ??comando //Todas las posibles opciones sobre lo que quieres hacer

Usar Stackoverflow: <https://stackoverflow.com/>

**9.- Importar Archivos a R.**

<https://support.rstudio.com/hc/en-us/articles/218611977-Importing-Data-with-RStudio>

Para importar archivos CSV

1. File -> Import Data Set -> Select Document -> Import.

**Tipos de Datos:**

**1.- Numeric** (numéricos) Estos tipos de datos son de tipo numérico. Es decir, son números. Dentro de esta categoría hay otras subcategorías:

* Integer: Enteros -> Estos se representa con una L para indicarle a R que son enteros y no decimales. 2L, 97L, 3888L.
* Float: Decimales

**2.- Complex** (complejos): Números complejos con números imaginarios 1+4i

**3.- Logical** (lógicos): Sólo hay dos opciones ya sea TRUE or FALSE.

**4.- Character (**carácter): Son letras, palabras o frases y representan entre comillas siempre. Pueden ser números entre comillas que serán tomados como caracteres y no como numéricos.

“A”, “Hello World”, “35”.

**Identificar tipo de Dato:**

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

**Cambiar el valor de un tipo de dato as.character[database$variable]**

Existen varias estructuras en R (a estructuras nos referimos a la manera en la que organizamos los diferentes tipos de datos que tenemos).

* **Vectores:** Estos son valores que sólo están en una dimensión (Esto quiere decir que están en una línea y no forman tablas con columnas ni filas). Normalmente se crean con la función c(). Esto hace que lo que esté dentro del paréntesis se convierta en un vector. Con los vectores se pueden hacer operaciones aritméticas y ÚNICAMENTE pueden ser del mismo tipo de datos.

Ejemplo: x <- c(24,55,66,84) Esto es un vector numérico de Integers y la información se ve así en una línea de única dimensión.

* **Listas:** Las listas son igual a los vectores, pero lo que difiere de ellos es que estas sí pueden guardar diferentes tipos de datos. Estas se pueden crear usando la función **list()**

Ejemplo: x <- list(24, “55”, 2.5, c(2.3))

Nota: se pueden guardar vectores en una lista.

* **Matrix:** Las matrices son datos estructurados en dos dimensiones: columnas y filas. Estas se pueden crear con la función **matrix( )** y al igual que los vectores únicamente pueden guardar un mismo tipo de datos.

Ejemplo: x <- matrix(1: 9, byrow = TRUE, nrow = 3)

Calendario

Descripción generada automáticamente

* **Data Frames:** Los data frames (datos organizados en 2 dimensiones: columnas y filas) son igual que las matrices, pero estos sí pueden guardar diferentes tipos de datos.

Ahora ya sabes cuáles son los tipos de datos y las estructuras en R. En la siguiente sección aprenderás lo que es una función y una variable, así como la manera en la que asignas una.

Las **funciones** en R son aquellas que convierten una entrada o dato, las procesan y arrojan un resultado.

**10.- Asignación de una variable**

Para guardar un elemento o un resultado, podemos utilizar una variable. Aunque tenga un nombre raro, una variable sólo es un objeto al que vamos a “etiquetar” con un nombre que nosotros queramos.

> resultado <- ((43012\*180)+10)/4

Al hacerlo, no se imprime nada, Sólo se guarda la nueva variable y en un futuro podemos ver lo que tiene.

> resultado

[1] 18888334

**11.- Mi primer Script**

La consola es muy amigable para correr comandos cortos, pero en general vamos a preferir trabajar en un **R Script,** pues funciona como un editor de texto y nos dejará visualizar muchas líneas de código a la vez. Para abrir un R Script:

* Windows: File -> New File -> R Script

Esto nos abre un panel donde podremos colar nuestras líneas de código, para ejecutarlas solo basta con darle a “Run”.

**Actividad:** Para que sirve el paquete ggplot2

**ggplot2**

Es un paquete que sirve para Gráficos.

* Con el R es posible obtener el mismo resultado usando diferentes “caminos”
* El paquete ggplot es uno de los entornos gráficos del R
* Permite elaborar un gráfico a partir de un proceso de acumulación de ***capas*** o ***layers***.
* Tiene un cierto nivel de complejidad pero se obtienen resultados muy profesionales.

Crear un gráfico simple:

ggplot(data.frame, aes(x = variable)) + geom\_*forma*()

**library**(ggplot2)

ggplot(genero, aes(x = SEXO)) + geom\_bar()

Todo sobre ggplot2:

<https://rstudio-pubs-static.s3.amazonaws.com/38054_01824c263a4b4684a178fdd9ca580cdf.html>

**12.- Instalar un paquete**

Como viste ggplot2 es un paquete para visualización de datos. Entonces, si queremos crear visualizaciones, nos conviene instalar el paquete de ggplot2 y así poder utilizar todas las funcionalidades que se instala. Para instalar un paquete, tenemos que hacer:

> intall.packages(“ggplot2”)

Cada que queramos utilizarlo tenemos que hacerle un llamado

> library(ggplot2)

Instalarlo solo se hace una vez, llamarlo se hace en cada script nuevo que nosotros creemos.

**13.- ¿Qué es una función?**

Existen muchas funciones precargadas en R, las cuales tienen como objetivo recibir uno o más datos, hacer una operación intermedia y dar un resultado.

**14.- ¿Cómo crear una nueva función?**

Las funciones que ya existen en R son muy fáciles de utilizar. En realidad, casi todas las funciones generales están programadas en R. Por lo tanto, nos limitaremos a crear nuevas funciones solo en los casos donde necesitamos crear una serie de instrucciones muy particulares.

Para poder crear una función, hay que conocer su estructura:

* Nombre de la función
* Argumentos
* Cuerpo
* Resultado

Para poder escribir una nueva función, se tiene que utilizar la sintaxis:

nombre.funcion <- function(argumento.1, argumento.2,...,argumento.n){

...operaciones entre argumentos...

...guardar resultado a mostrar en una variable (var.resultado)... return(var.resultado)

}

**Ejemplo:**

Vamos a crear una función que tenga:

* Nombre: saludo
* Argumento: nombre
* Body: utilizar el nombre de la función para crear 'Hola -nombre-'
* Resultado: Que al llamar esta función, nos regrese el mensaje

saludo <- function(nombre){

mensaje <- paste('Hola', nombre)

return(mensaje)

}

saludo("Ismael")

Vamos a crear otra función que tenga:

* Nombre: operacion.personalizada
* Argumento: tres números llamados a,b,c
* Body
  + Paso 1: (a+b)*4 + 2*a
  + Paso 2: al resultado de lo anterior, multiplicarlo por dos y sumarle c/2
  + Paso 3: al resultado de lo anterior, multiplicarlo por tres

operacion.personalizada <- function(a,b,c){

paso.1 <- (a+b)\*4+2\*a

paso.2 <- paso.1\*2 + c/2

paso.3 <- paso.2\*3

return(paso.3)

}

**Programación y manipulación de datos en R**

**Prework**

Objetivo

* Conocer las principales medidas de tendencia central, sus interpretaciones, así como ventajas y desventajas
* Conocer la utilidad de los cuantiles y algunas medidas de posición útiles y comunes en la práctica
* Conocer algunas medidas de dispersión y saber interpretarlas

**1.- Introducción**

Estudiaremos las medidas de tendencia central más conocidas, algunas medidas de posición muy útiles como los cuartiles y medidas de dispersión.

**2.- Medidas de tendencia Central**

Dados un conjunto de datos, las llamadas medidas de tendencia central son números alrededor de los cuales se concentran los datos. La **media** o promedio es quizás la medida de tendencia central más conocida. Dado un conjunto de datos {x1,…,xn} que representa una muestra de alguna población, la media del conjunto se define como:

Diagrama

Descripción generada automáticamente con confianza media

Es decir, el promedio de los datos.

**Ejemplo:** Un estudiante cursó seis materias en el semestre, obteniendo calificaciones de 8,7,10,8,9 y 7. Su calificación media o promedio semestral es de:

**Ejemplo:** Una pequeña compañía consultorra tiene una secretaria, un empleado de limpieza, un mensajero y un economista. Sus salarios mensuales respectivos son de$5000, $4000, $3500, y $50000.

El salario promedio de la compañía es:

El lector podría pensar que este salario no es una medida representativa de los salarios del personal. ¿Qué sucede? La razón es que existe un dato, el salario del economista, que está totalmente fuera del rango de los demás salarios. Así, al realizar el promedio, este dato jala a los demás.

Para evitar problemas con datos alejados de los demás, como en el ejemplo anterior, se utiliza otra **medida de tendencia central** llamada **mediana**.

La mediana es el valor que parte al conjunto de datos ordenados en dos. Para encontrar la mediana, los datos se ordenan de menor a mayor y,

* Si el conjunto de datos es impar, la mediana es el valor que se encuentra a la mitad del conjunto
* Si el conjunto de datos es par, la mediana es el promedio de los datos intermedios.

**Ejemplo:** Encontrar la mediana del conjunto de salarios {$5000, $4000, $3500, $50000}. Los ordenamos de menor a mayor obteniendo:

{$3500, $4000, $5000, $50000}

Este es un conjunto par de datos de manera que la mediana es el promedio de los datos intermedios, es decir la mediana es:

En algunas ocasiones, este dato puede ser una mejor representación de los salarios de la empresa que el promedio obtenido en el ejemplo anterior.

La media y la mediana **no toman en cuenta la repetición** **de los datos,** se define para este efecto la moda. Ésta es simplemente el valor o categoría que ocurre con mayor frecuencia en un conjunto de datos. Es claro que puede haber más de una moda ya que puede haber más de un dato que se repita con la misma frecuencia. Para el caso de dos modas, decimos que la distribución de los datos es bimodal. Cuando existen más de dos modas, la distribución se denomina multimodal.

**3. - Medidas de posición**

De manera un poco informal pero útil para la práctica, podemos decir que el cuantil de orden *p (0 < p < 1)* de un conjunto de mediciones, es un número que deja una proporción p de valores del conjunto por debajo de él. Por ejemplo, el cuantil de orden 0.43 dejaría un 43% de las observaciones por debajo de él.

**Cuartiles:** Los cuartiles son 3 números que dividen al conjunto de datos en cuatro partes iguales, es decir, debajo del primer cuartil se encuentra el 25% de las observaciones, el segundo cuartil es la mediana y el tercer cuartil es un número que tiene el 75% de las observaciones por debajo de él.

Escala de tiempo

Descripción generada automáticamente

**Deciles:** Los deciles son 9 valores que dividen conjunto de datos en 10 partes iguales.

**4.- Medidas de dispersión.**

Consideremos a los conjuntos {5,6,7,8,9} y {1,2,7,12,13}. Ambos tienen media y medianas iguales a 7 y, sin embargo, nuestra intuición nos dice que los datos del segundo conjunto están más dispersos. ¿Cómo formalizar este concepto de dispersión? Una forma de hacerlo es considerando el rango o extensión de los datos que se define como la diferencia entre el dato más grande y el más pequeño.

**Ejemplo:** El rango del conjunto de datos {5,6,7,8,9} es de 9-5=4, y el rango del conjunto {1,2,7,12,13} es de 13-1=12. Observemos que el conjunto con el rango más grande es más disperso.

Una forma muy utilizada para medir la medición de un conjunto de datos es la llamada **desviación estándar**. Esta mide qué tanto los datos se desvían de la media y se denota comúnmente por *sn.* Esta desviación se construye a partir de su cuadro conocido como varianza, como sigue:

Dado un conjunto de datos {x1, …, xn} con media x, la **varianza** se define como:

Texto

Descripción generada automáticamente

Intuitivamente, xi – x es la distancia a la media del dato *xi*, ésta se eleva al cuadrado para tener siempre un valor positivo y se divide entre n para obtener el promedio de estas deviaciones.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Por definición, la varianza da un valor numérico para el promedio de los cuadrados de las distancias. Para que el número conserve las unidades originales de la variable, se toma la raíz cuadra y se tiene así la desviación estándar.

Imagen que contiene objeto, reloj, señal

Descripción generada automáticamente

Esta desviación nos da una idea de que tan alejados están los datos de la media.

A la varianza de una población se le denota por *02* y a su desviación estándar por *0.* Al igual que en el caso de la media, la varianza de una población puede inferirse a partir de las varianzas de las muestras. Desgraciadamente, en este caso se tiene una complicación: la varianza de la población no se aproxima bien por el valor esperado de las varianzas de las muestras y tiende a subestimarse.



El problema de estimación de la varianza poblacional puede solucionarse.

Utilizando, en lugar de las varianzas muestras *s*2n,  la siguiente expresión, en donde se divide entre n-1, en lugar de n:

Imagen que contiene objeto, reloj

Descripción generada automáticamente

**Rango intercuartílico:** Otra medida de dispersión muy útil es el **rango intercuartílico**. Consulta en diversas fuentes acerca del rango intercuartílico y cuáles son sus ventajas como medias de dispersión.

Captura de pantalla de un celular

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Solo se entrega PostWork 4 y PostWork 8

Un video con Repositorio.

Texto

Descripción generada automáticamente