UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Análisis exploratorio de datos con R project

MATERIAL DIDÁCTICO

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Jorge López Jiménez

ASESORA DE MATERIAL DIDÁCTICO

Gabriela Betzabé Lizárraga Ramírez

Ciudad Universitaria, Cd. Mx., 2023

Índice

Contexto del trabajo

Cantidades vastas de datos sobre diferentes aspectos de la vida son almacenadas en la nube. No solo datos de comportamiento e interacción humana a través del Internet son almacenados, también información financiera, médica, bioinformática, gubernamental, educativa, adquisiciones y la lista sigue. Lo interesante de estos datos no es sólo su magnitud, sino también la correlación que estos tienen entre sí mismos y el contexto del cual son extrapolados.

En muchas ocasiones, estos datos acaban convirtiéndose en piezas de un producto, el cual retroalimenta a la persona con la que se le relacionan estos datos; por ejemplo, Amazon con su sistema de recomendación de productos o Facebook con su sistema de recomendación de amistades.[1]

El cómputo estadístico es el área de estudio que surge de la interacción entre la Estadística y la Ciencia de la Computación, usando conceptos de ambas ramas para generar herramientas y modelos para descubrir relaciones entre los datos y generar contexto.

Objetivo de la propuesta

El cómputo estadístico es el área de estudio que surge de la interacción entre la estadística y la ciencia de la computación. Esta vertiente ha evolucionado rápidamente, por lo que cada vez se enseñan conceptos de computación más complejos dentro del área de la estadística. La presente serie de prácticas son un método introductorio para que el estudiante se involucre con las herramientas que se usan en el análisis exploratorio de datos y los tipos de análisis que regularmente se piden en las reglas de negocio, que potencialmente puede

[1] Doing Data Science by Rachel Schutt and Cathy O'Neil 2014, pag. 4.

encontrar a lo largo de su carrera profesional como Ingeniero en Computación especializado en datos, o como a lo largo de las prácticas se referirá: “Ingeniero de datos”.

El objetivo de este compendio de prácticas es ampliar y actualizar los conocimientos de la materia de Minería de Datos a través de cinco prácticas que abordan tópicos esenciales para análisis de grandes cantidades de datos, popularmente conocido en la industria como macrodatos o datos masivos.

Dichas prácticas usan software de licencia libre y serían parte complementaria del curso. Los requerimientos para realizar estas prácticas son cubiertos con cualquier tipo de computadora personal, ya que para fines académicos los conjuntos de datos que se usan en las prácticas no caben en la categoría de macrodatos pero son perfectamente trasladables a los mismos.

Este compendio guía al alumno paso a paso en el desarrollo de las cinco prácticas y está pensado para ser desarrollado en el tiempo de estudio designado a la materia.

Al finalizar las prácticas, el profesor podrá corroborar el nivel de aprendizaje alcanzado por los alumnos con los ejercicios propuestos en cada una de ellas.

Definición del problema

En la actualidad, los datos son considerados como uno de los activos más importantes para cualquier empresa. Tal como hace cien años el petróleo fue la base de la industria energética y era crucial tener profesionales altamente capacitados para su extracción, procesamiento y transporte; en la actualidad, es imprescindible que los ingenieros de datos sean expertos en la extracción, transformación y carga de los mismos. Justo ahora existen herramientas que ayudan a los expertos en datos con la ejecución de las principales tareas de su

oficio. Estas nuevas tecnologías son lo que permite darles valor a los mismos y convertirlos en información.

El acercamiento tradicional de los ingenieros de datos suele ser usar manejadores de bases de datos relacionales con SQL como lenguaje principal para manipular los datos, sin embargo, hay ciertos sectores de la industria que generan datos más complejos de los que pudiese analizar con eficiencia un equipo de cómputo promedio con una base de datos relacional y el lenguaje SQL.

IBM, Amazon y otras grandes compañías informáticas han creado herramientas para el análisis de datos complejos y heterogéneos (SAS, Tableau, Power BI, entre otros) y también existen herramientas bajo la licencia de software libre para realizar estas tareas, como R Project[2], Pandas[3] o NumPy[4].

En estas prácticas se emplea R Project, que es un entorno de software libre para la computación estadística y la creación de gráficos. Dicho lenguaje compila y corre en sistemas Windows, Unix, Linux y MacOS.

Según el Foro Económico Mundial, en los próximos cuatro años habrá una adopción generalizada de las tecnologías de analítica de datos a gran escala, puesto que tendrán un mayor impacto en los empleos del futuro por la llegada de las nuevas redes de comunicación 5G, la inteligencia artificial, la adopción generalizada los macrodatos y el poder del cómputo en la nube generarán cambios en toda la sociedad.[5] Por lo tanto, es importante dirigir a los estudiantes a emplear herramientas de software libre para que adquieran la habilidad de identificar, tratar y emplear todos los recursos que estén a su alcance de manera eficaz. Además, que su participación como expertos de datos contribuya al

[2] About R by R Foundation. Obtenido de About R: https://www.r-project.org/about.html.
[3] About Pandas by Pandas community. Obtenido de About Pandas: https://pandas.pydata.org/about/index.html.
[4] About NumPy by NumPy community. Obtenido de About NumPy: https://numpy.org/#.
[5] Big Data, Big Impact : New Possibilities for International Development https://www.weforum.org/reports/big-data-big-impact-new-possibilities-international-development

tratamiento de los mismos en las empresas mexicanas y que estas habilidades los conviertan en profesionales competitivos a nivel nacional e internacional.[6]

En la propuesta que se presenta, mediante cinco prácticas, se explican los conceptos básicos y se guía al alumno por estos temas para su conocimiento y futuro desarrollo con la ingeniería de datos.

Estructura

Los temas propuestos son:

- EDA
- Gráficas con ggplot2
- Análisis exploratorio de datos espaciales con R
- Virtualización de datos
- Regresión lineal

Estos temas llevarán al estudiante a reconocer la naturaleza de los datos y el empleo de las herramientas para su limpieza y normalización. Una vez que cuenten con la calidad del dato que se requiere, se le mostrará cómo explorarlos, así pudiendo hacer sus propias hipótesis bajo métodos estadísticos. Esta exploración se hace con ayuda del lenguaje R.

Cada uno de los comandos se mostrará a lo largo de las prácticas como casos de estudio reales; así el alumno no solo aprenderá algoritmos, también se enfrentará y resolverá los problemas técnicos más comunes presentados en el día a día de un ingeniero de datos.

[6] La ciencia de datos gana terreno en la educación superior mexicana by Antonio Becerril. Obtenido de El Economista: https://www.eleconomista.com.mx/tecnologia/La-ciencia-de-datos-gana-terreno-en-la-educacion-superior-mexicana-20190414-0008.html.

La introducción general tiene la finalidad de situar al alumno en el contexto actual del estudio y análisis de datos, brindándole la terminología correcta y motivación para estudiar ingeniería de datos.

La primera práctica enseñará al alumno a explorar los datos que se quieren analizar, a través de comandos y herramientas matemáticas.

El objetivo principal de este acercamiento es que se exploren los datos y se dé una perspectiva general de estos.

En la práctica dos, se brindarán las primeras herramientas que se tienen como ingeniero de datos. Esta serie de algoritmos serán la base a la cual el alumno recurrirá para obtener valores de los datos, desde correlaciones hasta tendencias.

Dentro de cada uno de los temas emplearán modelos matemáticos para predicción y filtrado. Para estos temas se requerirá que el alumno cubra con los conocimientos en matemáticas, estipulados en el apartado de requerimientos.

En cada práctica se ven casos de estudio, donde la tarea de un ingeniero de datos se hace presente, y se mencionan los casos particulares en que se usa ciencia de datos y cómo se resuelven los desafíos presentados.

La ciencia de datos no es solo un concepto sintético para unificar estadísticas, análisis de datos y sus métodos relacionados, sino que también comprende sus resultados. Incluye tres fases, diseño de datos, recopilación de datos y análisis de datos. Los conceptos fundamentales y varios métodos basados en ellos se discuten con un ejemplo heurístico. (Chikio Hayashi, p.55)[7]

En la práctica tres se le proveerá de librerías y arquitecturas específicas al alumno para que pueda generar mapas temáticos y usarlos como referencia a los datos que se están procesando y analizando.

En tanto, en la práctica cuatro aprenderá cómo crear librerías de datos semi aleatorizadas.

[7] Data Science, Classification, and Related Methods, Chikio Hayashi · Keiji Yajima Hans H. Bock · Noboru Ohsumi Yutaka Tanaka · Yasumasa Baba, 1996

Para finalizar, en la práctica cinco, el alumno aprenderá a verificar los análisis realizados mediante el empleo de la retroalimentación inmediata con base en la regresión lineal. Así, el alumno podrá utilizar los datos generados previamente como parte de un proceso comprobatorio que le permitirá verificar la estrategia utilizada con anterioridad y nutrir su manejo de los datos.

Justificación

Introducir al alumno al análisis de datos a través del Lenguaje de programación R.

Aunque es cierto que el tema de análisis de datos o minería de datos no es algo nuevo dentro de la carrera de ingeniería de computación, también es cierto que la industria de los datos sigue creciendo exponencialmente.[8] Dado a este crecimiento y la necesidad de profesionales que ayudan en el análisis de datos se creó esta serie de prácticas. El objetivo es que el alumno siga paso a paso cada una de las prácticas y al final de cada una de ellas, ponga a prueba lo que practicó.

Requerimientos

A lo largo de esta opción de titulación se asume que el alumno conoce de:

- Álgebra lineal
- Probabilidad
- Estadística
- Programación funcional
- Programación a nivel de sistema operativo shell scripting.

El material a emplear será un equipo de cómputo con conexión a Internet.

[8] https://www.fortunebusinessinsights.com/industry-reports/big-data-technology-market-100144

Método

Método deductivo directo – inferencia o conclusión inmediata

Se emplearán como artefacto de enseñanza cinco prácticas estructuradas de la siguiente manera:

1. **Introducción**

Breve explicación sobre el tema para dar al alumno un acercamiento teórico a la herramienta que se usará y justificación de porqué esa herramienta es utilizada en el manejo y exploración de los datos.

2. **Requerimientos**

Se pedirá al alumno preparar el ambiente de trabajo llevándolo de la mano para construir, en su propia computadora, un laboratorio de manejo y análisis de datos. Se usarán datos que representen, para él, un inicio en el manejo de datos masivos mediante la extrapolación a bases grandes de datos de las habilidades adquiridas.

3. **Desarrollo**

Paso a paso se pedirá al alumno, mediante capturas de pantalla y explicaciones cortas, que vaya alcanzando los diferentes objetivos planteados en cada práctica.

4. **Conclusión**

Dentro de esta sección se explicará los detalles finales de lo que se hizo y por qué se hizo así, a su vez se le planteará un escenario distinto, para el cual pueda usar los conocimientos adquiridos dentro del desarrollo.

5. **Ejercicio**

Una vez aterrizados los conceptos de la práctica, se menciona otra posible área de oportunidad para evidenciar los saberes adquiridos por el alumno empleando sus nuevos conocimientos para complementar estos requerimientos formales.

Cronograma de actividades

Se sugiere el desarrollo de estas prácticas durante las cinco semanas subsecuentes acorde a la siguiente tabla:

Práctica/Semana	1	2	3	4	5
1. EDA					
2. Gráficas con ggplot2					
3. Análisis exploratorio de datos espaciales con R					
4: Virtualización de datos					
5. Regresión lineal					

Práctica 1. EDA

Introducción

El **análisis de datos** es el proceso de inspeccionar, limpiar y transformar datos con el objetivo de resaltar información útil; para, con esto, establecer conclusiones y ser un sustento de apoyo a la toma de decisiones. Tiene múltiples facetas y enfoques en diversas técnicas aplicables a los negocios y las ciencias.

Objetivo

El objetivo de esta práctica es explorar una base de datos proporcionada por el INEGI, referente a la medición de progreso de las sociedades surgido del consenso internacional por la Comisión Sobre la Medición del Desempeño Económico y el Progreso Social (Comisión Stiglitz-Sen-Fitoussi).

Esta base de datos está apegada a las recomendaciones para captar estadísticamente el Bienestar Subjetivo, formuladas por la Organización para la Cooperación y el Desarrollo Económico (OCDE) de la que nuestro país es miembro.

El bienestar subjetivo hace referencia a las experiencias de vida en primera persona, por lo cual es indispensable analizar con este muestreo cómo están experimentando su vida los mexicanos.[9]

[9] Instituto Nacional de Estadística y Geografía . (Abril de 2017). Obtenido de Bienestar subjetivo: https://www.inegi.org.mx/proyectos/investigacion/bienestar/basico/default.html

Requerimientos

Primero, se descarga la herramienta requerida para hacer la exploración de los datos. R es un lenguaje de programación para el cómputo estadístico y generación de gráficos. Es un proyecto bajo la licencia GNU, el cual es muy similar al lenguaje S que fue desarrollado en los laboratorios Bell por John Chambers y sus colegas. Es un grupo de herramientas de software para la manipulación de datos, cálculo y generación de gráficos.[10] Inicialmente fue pensado como un ambiente de desarrollo para cómputo estadístico; sin embargo, la comunidad de ingenieros y científicos han extendido las capacidades de R por medio de paquetes de funciones, al grado de verse usado tanto como Python para cómputo de alto rendimiento y para aprendizaje de máquinas.

1. Instalación de R

R se puede instalar en cualquier servidor tipo UNIX, (Linux) y Windows de manera gratuita desde el sitio oficial https://cran.r-project.org/mirrors.html

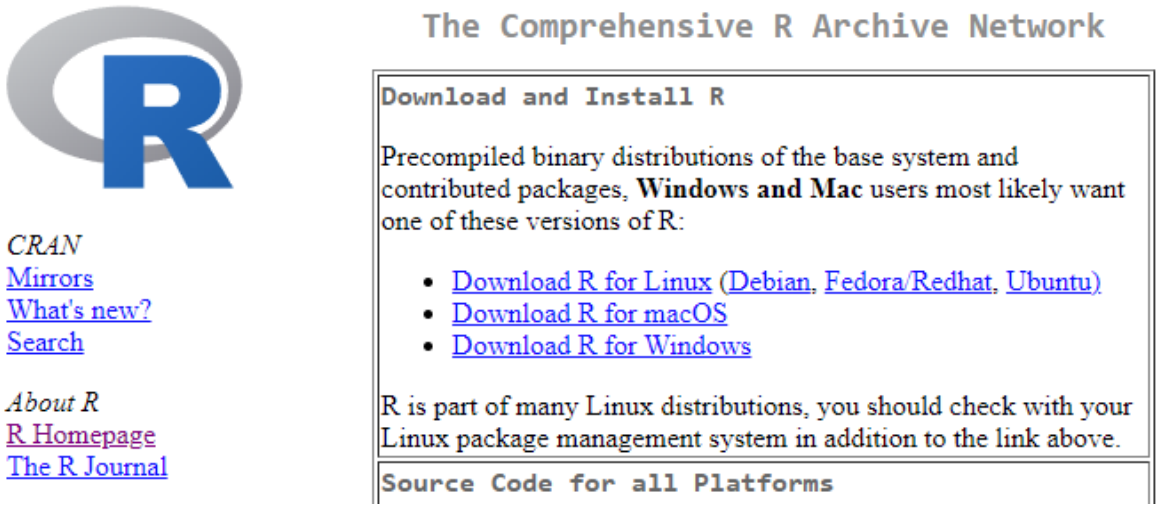


Figura 1. Sitio de descarga de R

[10] https://www.r-project.org/about.html

Se elegirá la opción “Download R for…” según sea el sistema operativo para la descarga del ejecutable o de la paquetería.

Windows:

Se debe dar clic en el enlace “install R for the first time” para la instalación de R por primera vez en el equipo de cómputo.

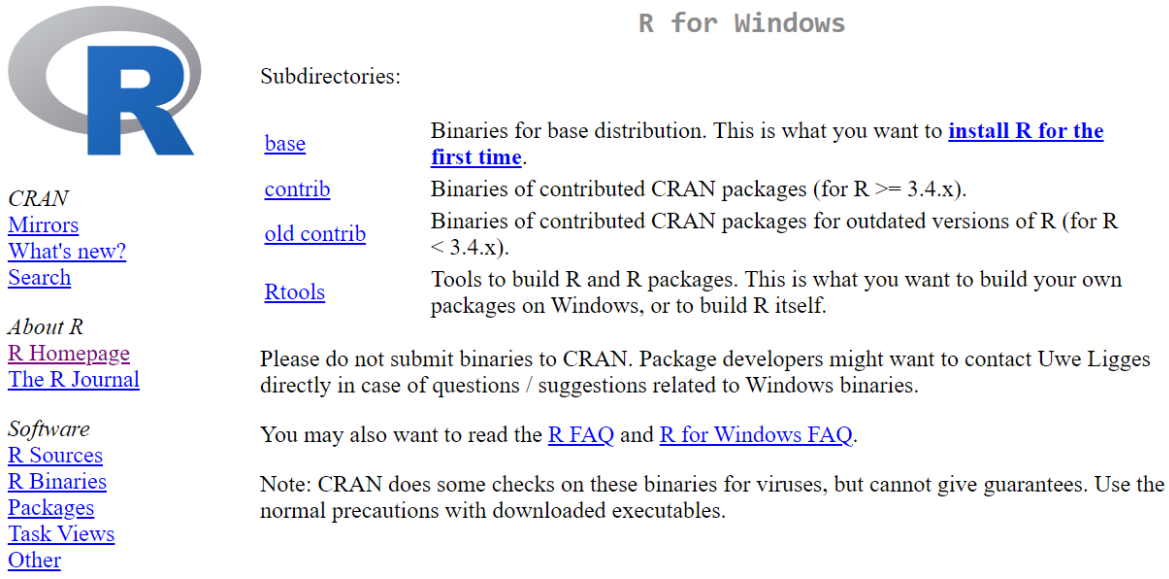


Figura 2. Sitio de descarga para windows

Linux:

Se seleccionará la opción que más se adapte a la distribución del sistema operativo. Dependiendo de la distribución de Linux se debe elegir el manejador de paquetes adecuado.

Debian

```
$ apt-get update
$ apt-get install r-base r-base-dev
```

Fedora/Redhat

```
$ sudo dnf install R
```

Ubuntu

```
$ apt update -qq

$ apt install --no-install-recommends software-properties-common dirmngr

$ wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc | sudo tee -a /etc/apt/trusted.gpg.d/cran_ubuntu_key.asc

$ add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu $(lsb_release -cs)-cran40/"

$ apt install --no-install-recommends r-base
```

De igual manera se puede descargar directamente el archivo binario para compilarlo.

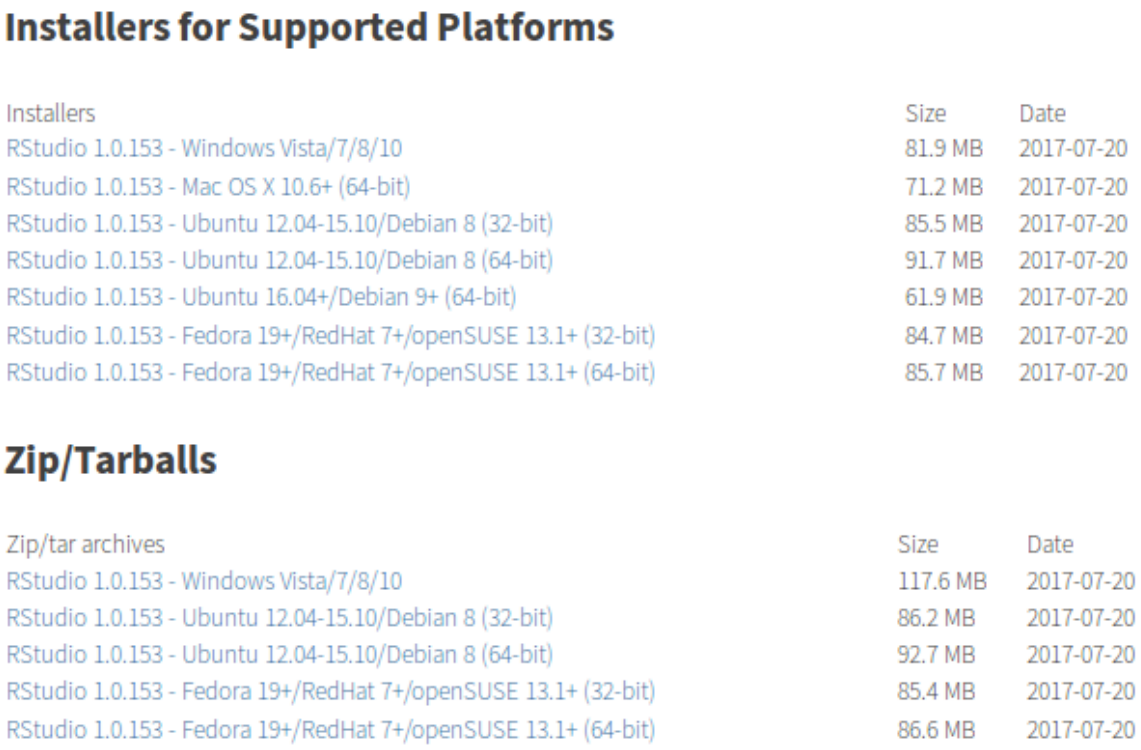
2. Instalación de RStudio

El siguiente paso será la instalación del ambiente gráfico RStudio, el cual es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para R; este, incluye una consola, un editor con formateo dinámico según la sintaxis del lenguaje que soporta ejecución del código directamente, herramientas para gráficas, historial, depuración y manejo del entorno gráfico.

RStudio tiene versiones open source y comercial. Se usará la versión open source para cada una de las prácticas.

Para su descarga, se usará el siguiente enlace: https://www.rstudio.com/products/rstudio/download/ (R Studio, 2017).

En la parte inferior del sitio web se encuentran los enlaces a la descarga de la aplicación, igual que con el interpretador de R, se elegirá la que más se adapte al sistema operativo que se emplee.



Installers for Supported Platforms

Installers	Size	Date
RStudio 1.0.153 - Windows Vista/7/8/10	81.9 MB	2017-07-20
RStudio 1.0.153 - Mac OS X 10.6+ (64-bit)	71.2 MB	2017-07-20
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	85.5 MB	2017-07-20
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	91.7 MB	2017-07-20
RStudio 1.0.153 - Ubuntu 16.04+/Debian 9+ (64-bit)	61.9 MB	2017-07-20
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	84.7 MB	2017-07-20
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	85.7 MB	2017-07-20

Zip/Tarballs

Zip/tar archives	Size	Date
RStudio 1.0.153 - Windows Vista/7/8/10	117.6 MB	2017-07-20
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	86.2 MB	2017-07-20
RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	92.7 MB	2017-07-20
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	85.4 MB	2017-07-20
RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	86.6 MB	2017-07-20

Figura 3. Opciones de descarga por sistema operativo

Una vez descargado, se procederá a la instalación de manera local en la máquina.

Desarrollo

1. Reglas del negocio

Una vez instalado el ambiente de desarrollo del lenguaje R, se procederá a describir las reglas de negocio, el objetivo de un científico de datos siempre es responder preguntas y generar información a partir de los datos, así que se practicará alrededor de la pregunta:

¿Qué tan satisfechos con su vida se perciben los mexicanos actualmente?

BIARE es la contracción de Bienestar Autorreportado, *bienestar* porque se pretende captar la situación en la que se encuentran las personas y a*utorreportado* porque es la misma persona, desde su propia percepción, la que

nos dice cómo ve y valora su situación (bienestar) independientemente del tipo de encuesta que se esté levantando: cara a cara o autollenado.

Disponer de información que refleje el grado o nivel de bienestar de la población mexicana es un reto que el INEGI está haciendo posible con el levantamiento de una encuesta, que pone en contacto al informante consigo mismo, con sus sentimientos y vivencias; los cuales, aunque sean positivos o negativos, influyen en el progreso de la sociedad. De hecho, el bienestar autorreportado o bienestar subjetivo es uno de los elementos fundamentales dentro de lo que se ha denominado como “medición del progreso de las sociedades” que es una iniciativa adoptada primeramente por los gobiernos de Francia y Reino Unido, y, posteriormente, por la Organización para la Cooperación y el Desarrollo Económico (OCDE). Por lo tanto, el BIARE cabe dentro de las encuestas llamadas de percepción, que cada vez están cobrando más relevancia en el mundo como estadística oficial.

Para descargar se tiene disponible el siguiente enlace del INEGI:

https://www.inegi.org.mx/investigacion/bienestar/basico/default.html

(Instituto Nacional de Estadística y Geografía , 2017)

Descargar 3 archivos:

- De la sección *Microdatos*, la base de datos más reciente en formato CSV.
- De la sección *Microdatos*, El documento en formato PDF *Estructura de la base de datos* el cuál contiene la descripción de las 3 partes de la encuesta, así como la descripción de cada campo de la misma titulado biare_fd_2015.pdf
- De la sección *Documentación,* El archivo en formato PDF *Instructivo de llenado del cuestionario* el cuál contiene el instructivo del cuestionario y descripción de ciertos campos importantes.

2. Cargar archivo CSV

Se abre RStudio, el cual será nuestra interfaz gráfica.

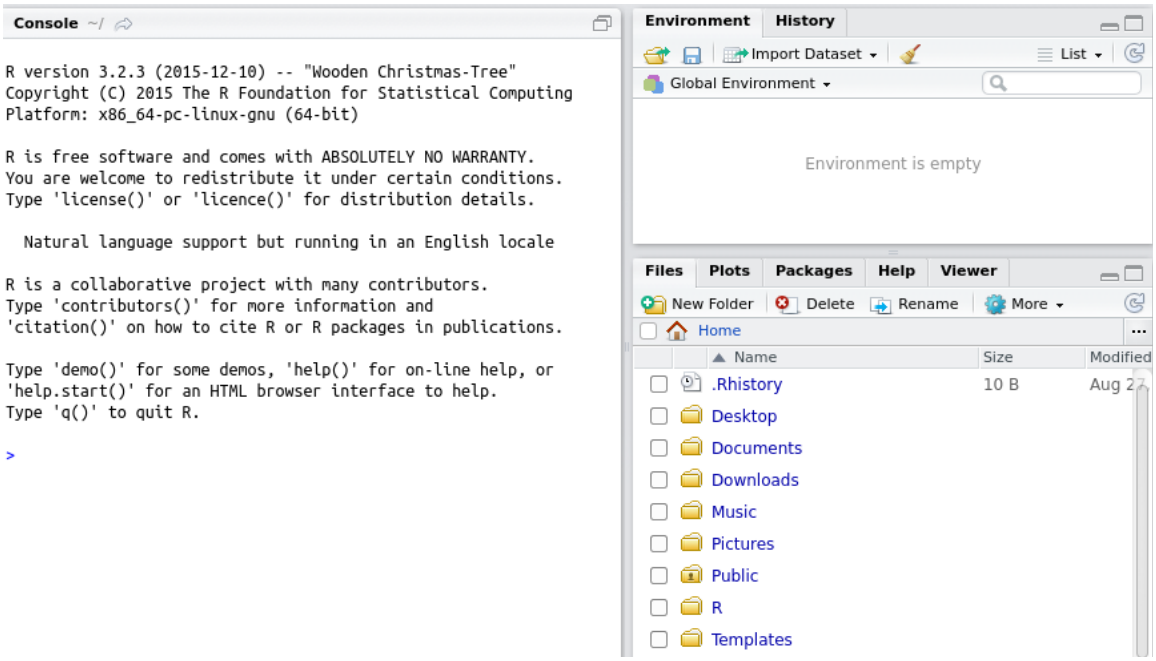


Figura 4. Primera pantalla de la interfaz gráfica de RStudio

En esta interfaz se encontrarán 3 ventanas principales;

- **Console**
  - Es la consola interactiva de R, la cual permitirá escribir el código que se requiera nuestro interpretador reciba y ejecute al vuelo.
- **Environment/History**
  - Aquí se encuentra un listado de los objetos que se van creando en la sesión activa de R (variables, data frames, funciones etc.), en la otra pestaña se encuentra el historial de todas las instrucciones que se ejecutan en consola.
- **Files/Plots/packages/Help/Viewer**
  - Explorador de archivos genérico.

- Vista previa de las gráficas.
- Paqueterías cargadas.
- Documentación de las paqueterías.
- Un visor de contenido web para paqueterías de R.

Una vez familiarizados con el ambiente gráfico, se procede a cargar los datos en un data frame. *Un data frame es una estructura de datos usada para almacenar datos en forma de tablas, en código es una lista de vectores de un mismo tamaño.*

Para esto, se deben tener previamente los archivos que se descargaron del enlace del sitio del INEGI en una carpeta descomprimidos. Se carga el archivo 'biare_cb' usando la función read.csv, a la cual se le pasa como argumento la ruta de éste.

Para dar un vistazo general a los datos se usa la función head. La cual recibe como argumento el objeto que se quiere visualizar, (al igual que opcionalmente el número de filas por observar) en este caso es el nuevo objeto que contiene la fuente de datos.

```
> biare_cb <- read.csv('<<ruta absoluta a "biare_cb_xxxx_xx.csv">>')

> head(biare_cb, 3)
```

Como resultado se muestra un extracto de la matriz de datos, tal como se observa en la Figura 5:

```
> head(biare_cb, 3)
  PER N_ENT    FOL_ENT   CON V_SEL N_HOG H_MUD N_REN P1 P2 P3_1 P3_2 P3_3 P3_4 P3_5 P3_6
1 122     3 12A207   1 40007     2     1     0     2  6  5   10   10   10   10   10   10
2 122     3 12A207   1 40007     3     1     0     2  9  9   10   10   10   10   10   10
3 122     3 12A207   1 40007     4     1     0     1  8  8   10   10   10   10   10   10
  P3_7 P3_8 P3_9 P3_10 P3_11 P4_1 P4_2 P4_3 P4_4 P4_5 P4_6 P4_7 P4_8 P4_9 P4_10 P5_1 P5_2
1   10   10   10    10     5   10   10   10   10   10    0    0    0    0     0   10   10
2   10   10   10     7     3   10   10   10   10   10    0    0    0    0     0   10    8
3   10   10   10    10    10    7   10   10    8    8    3    0    0    2     2   10    8
  P5_3 P5_4 P5_5 P5_6 P5_7 P5_8 P5_9 P5_10 P5_11 P5_12 TIPO FAC_MOD
1   10    5   10   10   10   10   10    10    10    10    1   11900
2    8   10    7   10    0   10   10     8     8     7    1   23801
3   10   10   10    8   10   10   10    10    10    10    1   23801
```

Figura 5. Resultado de la función head()

El archivo titulado *Estructura de la base de datos* (biare_fd_2015.pdf) contiene la descripción de los campos relacionados a los datos del encuestado y su hogar.

Base: **BIARE_CB.DBF**

Campo	Mnemónico	Tipo	Longitud	Códigos Válidos	Descripción
1	PER	Character	4	M 01-12, A 07*	Periodo de la entrevista
2	N_ENT	Character	1	1 - 8	Número de la entrevista
3	FOL	Character	6	1 Zona, 1 - 3 Estrato, Últimos 4 panel de rotación	Folio
4	ENT	Character	2	01-32	Entidad
5	CON	Character	5	00001-99999	Control
6	V_SEL	Character	1	1 - 4	Vivienda seleccionada
7	N_HOG	Character	1	1	Número de hogar
8	H_MUD	Character	1	0 - 7	Hogar mudado
9	N_REN	Character	2	01-30	Número de renglón
10	P1	Numeric	2	1-2	1. ¿Podría decirme en una escala de 0 a 10 qué tan satisfecho se encuentra actualmente con su vida?
11	P2	Numeric	2	1-2	2. Y hace un año, ¿qué tan satisfecho se encontraba con su vida?
	P3			En una escala de 0 a 10 qué tan de acuerdo o desacuerdo está usted con la frase…	
12	P3_1	Numeric	2	0-10	1. En general me siento bien con respecto a mí mismo
13	P3_2	Numeric	2	0-10	2. Siempre soy optimista con respecto a mi futuro
14	P3_3	Numeric	2	0-10	3. Soy libre para decidir mi propia vida
15	P3_4	Numeric	2	0-10	4. Tengo fortaleza frente a las adversidades
16	P3_5	Numeric	2	0-10	5. Por lo general siento que lo que hago en mi vida vale la pena
17	P3_6	Numeric	2	0-10	6. Soy una persona afortunada
18	P3_7	Numeric	2	0-10	7. El que me vaya bien o mal depende fundamentalmente de mí
19	P3_8	Numeric	2	0-10	8. Siento que tengo un propósito o una misión en la vida
20	P3_9	Numeric	2	0-10	9. La religión es importante en mi vida
21	P3_10	Numeric	2	0-10	10. La mayoría de los días siento que he logrado algo
22	P3_11	Numeric	2	0-10	11. Cuando algo me hace sentir mal me cuesta mucho volver a la normalidad
	P4			Las preguntas que le voy a hacer a continuación se refieren a ¿qué tanta parte del día de ayer se sintió…	
23	P4_1	Numeric	2	0-10	1. … de buen humor?
24	P4_2	Numeric	2	0-10	2. … tranquilo, calmado o sosegado?
25	P4_3	Numeric	2	0-10	3. … con energía o vitalidad?
26	P4_4	Numeric	2	0-10	4. … concentrado o enfocado en lo que hacía?
27	P4_5	Numeric	2	0-10	5. … emocionado o alegre?
28	P4_6	Numeric	2	0-10	6. … de mal humor?
29	P4_7	Numeric	2	0-10	7. … preocupado, ansioso o estresado?
30	P4_8	Numeric	2	0-10	8. … cansado o sin vitalidad?
31	P4_9	Numeric	2	0-10	9. … aburrido o sin interés en lo que estaba haciendo?
32	P4_10	Numeric	2	0-10	10. … triste, deprimido o abatido?

Nótese que las cabeceras de la base de datos coinciden con la estructura descrita en el PDF, coinciden nombres, tipo de datos y su tamaño. Este es el primer paso en la preparación y limpieza de los datos, se tiene que revisar que el diccionario de datos contenga los campos y preferentemente una descripción de los mismos.

3. Limpiar datos.

Dentro del diccionario de datos se encontró que hay 4 columnas que no quedan totalmente descritas.

3	FOL	Character	6	1 Zona, 1 - 3 Estrato, Últimos 4 panel de rotación	Folio
4	ENT	Character	2	01-32	Entidad
5	CON	Character	5	00001-99999	Control
6	V_SEL	Character	1	1 - 4	Vivienda seleccionada
7	N_HOG	Character	1	1	Número de hogar
8	H_MUD	Character	1	0 - 7	Hogar mudado
9	N_REN	Character	2	01-30	Número de renglón

Sin embargo, podemos suponer que el tercer campo es un folio de control generado para cada entrevista hecha, así que se descartó. El cuarto campo refiere

a la entidad federativa en la cual se efectuó la encuesta, las claves para cada una de las entidades federativas son las siguientes:

Cve_ent	Entidad
01	Aguascalientes
02	Baja California
03	Baja California Sur
04	Campeche
05	Coahuila de Zaragoza
06	Colima
07	Chiapas
08	Chihuahua
09	Distrito Federal
10	Durango
11	Guanajuato
12	Guerrero
13	Hidalgo
14	Jalisco

Cve_ent	Entidad
15	México
16	Michoacán de Ocampo
17	Morelos
18	Nayarit
19	Nuevo León
20	Oaxaca
21	Puebla
22	Querétaro
23	Quintana Roo
24	San Luis Potosí
25	Sinaloa
26	Sonora
27	Tabasco
28	Tamaulipas
29	Tlaxcala
30	Veracruz de Ignacio de la Llave
31	Yucatán
32	Zacatecas

Por lo tanto, se conservará esa columna con el fin de organizar por estados los resultados de BIARE.

Del quinto al noveno campo, no se especifica naturaleza en el diccionario de datos. Así que se asume que todas estas variables son de identificación y no afectarán la naturaleza de los datos en sí; por lo tanto se eliminan los campos restantes.

```
> biare_cb[, c('FOL', 'CON', 'V_SEL', 'N_HOG', 'N_MUD', 'N_REN')] <-
NULL
```

Para referir a una variable, se emplea el operador '$'. El cual, al ser escrito inmediatamente después del nombre de la tabla, referirá a alguna de las columnas de ésta. Se comprobará la eliminación de las variables innecesarias en el data

frame. Esto se puede comprobar volviendo a traer los cabezales como se había hecho antes.

```
  per n_ent ent p1 p2 p3_1 p3_2 p3_3 p3_4 p3_5 p3_6 p3_7 p3_8 p3_9
1 717     3   1 10 10   10    9   10    8   10   10    9   10    9
2 717     3   1  8  8    9    9   10    8    8    9    7   10    7
3 717     3   1 10  8   10    9   10    9    9   10    9   10   10
4 717     3   1 10 10   10    8   10    9    9   10    7   10   10
5 717     7   1  7  5    8    8   10    7   10   10    5   10   10
6 717     7   1  9  9   10   10   10   10   10    9    8   10    8
  p3_10 p3_11 p4_1 p4_2 p4_3 p4_4 p4_5 p4_6 p4_7 p4_8 p4_9 p4_10
1     9     2    8    7    7   10    8    0    2    3    0     0
2     8     7    4    4    5    7    4    3    6    5    2     4
3     8     0    9   10    8   10    9    0    0    1    0     0
4     9     0    9    8    9   10    9    0    2    1    0     0
5    10     3    8    7    7    8    8    2    2    2    0     1
6    10     8   10    9    8    9   10    0    1    2    1     0
  p5_1 p5_2 p5_3 p5_4 p5_5 p5_6 p5_7 p5_8 p5_9 p5_10 p5_11 p5_12
1    9   10    9    9    8    8    5   10    9     8     9     8
2    7    5    7    8    5    4    2    7    8     8     8     5
3    8    8    9    9    8   10    4    9   10     9    10     7
4    8    7    8    9    8    9    2   10    8    10     9     9
5    7    7    8    8    8    5    8    8    8     8     8     8
6    9    9    9    9   10    8    8   10   10     9     9     8
  tipo fac_mod
1    1   15571
2    1   15571
3    1    7786
4    1   15571
5    1    8015
6    1   16029
```

Ahora, se instalarán las paqueterías “doBy” para facilitar el cómputo de datos por categorías. La librería doBy de R es una herramienta útil para el análisis y manipulación de datos. Te permite realizar operaciones de agregación, manipulación y análisis específicos en tus conjuntos de datos, lo que facilita la exploración y comprensión de la información contenida en ellos.

4. Instalando doBy

Una paquetería o librería es la unidad fundamental de código reproducible en R; es decir, estas paqueterías incluyen funciones reusables, documentación que las describe y datos de muestra.

Para su instalación se ejecutarán las siguientes instrucciones:

```
> install.packages("doBy")
> library("doBy")
```

Si después de la última instrucción no se arrojase ningún mensaje, sería indicio de que no se ha instalado correctamente la paquetería.

Lo que sigue será crear una función que tome como argumento un vector de datos de la siguiente manera

5. Funciones

Una de las características de R es poder generar funciones creadas por el usuario; de hecho, muchas de las funciones que se definen en R son funciones de funciones.

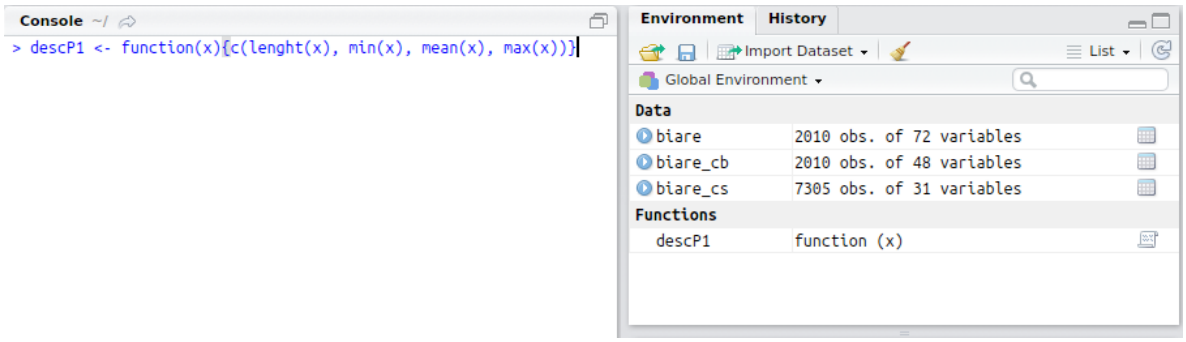
La estructura básica es la siguiente:

```
> myfunction <- function(arg1, arg2, ... ){
statements
return(object)
}
```

De esta manera, se construye una función propia a llamarse *descP1*. Se usa para describir alguna variable dentro del data frame. Se define como:

```
> descP1 <- function(x){c(lenght(x), min(x), mean(x), max(x))}
```

La función también es un objeto definido por el usuario dentro de la sesión; por lo tanto, se debería visualizar en la pestaña de *Environment*.



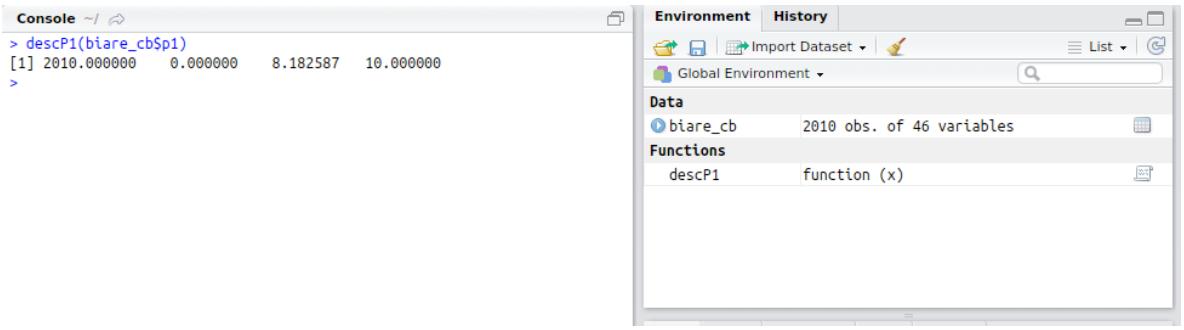
También, se puede apreciar en la imagen la definición de la función como el único comando que se ha ejecutado; al igual que una terminal de Linux, ésta se puede limpiar, simplemente se debe presionar CTRL + L.

Como se mencionó anteriormente, la mayoría de las funciones en R son funciones dentro de funciones. Esta que se acaba de definir no es una excepción.

La función propia está compuesta por cinco funciones:

1. c(...): Es una función genérica que combina sus argumentos para formar un vector.
2. Lenght
3. (x): Obtiene como argumento un vector, lista, factor o cualquier objeto en R para el cual su método haya sido definido.
4. min() y max(): Regresan el valor mínimo y máximo, respectivamente, de todos los valores presentes en sus argumentos; como entero, si estos argumentos son lógicos o enteros, o como doble si son todos del tipo numérico.
5. mean(): Función genérica para la media aritmética.

Por lo tanto, la función propia regresa un vector con los valores de la longitud, el mínimo, la mediana y el máximo valor de la variable del data frame que se le pasa como argumento.



Para la pregunta número uno, al encuestado se le da la opción de elegir un número del uno al diez para describir que tan satisfecho está con su vida. Respondiendo a la pregunta *"¿Podría decirme en una escala de 0 a 10 qué tan satisfecho se encuentra actualmente con su vida?"*

Muchas veces, se encuentran datos con escalas, porcentajes o valores continuos que, dada la naturaleza de éstos, se vuelve difuso y difícil extraer información a partir de su valor; en este tipo de casos es cuando conviene categorizar y etiquetar los resultados en grupos.

Se va a categorizar en grupos las posibles respuestas a esta pregunta con la reducción del rango a tres opciones.

```
> biare_cb$catP1 <- cut(biare_cb$p1, c(-Inf,3,7,Inf))
```

Aquí, se declararó una nueva variable dentro del data frame y se asignó el resultado de la función.

- cut(x, …): divide el rango del parámetro x en intervalos, toma cada valor de este parámetro y lo pone dentro del rango seleccionado. El segundo argumento sería vector con dos o más valores únicos, los cuales serán los puntos de corte del intervalo seleccionado. (Los valores *–Inf* y *Inf* representan los límites infinitos inferior y superior).

Por lo tanto, se usó la variable *p1* de nuestro data frame e hicimos que cayera dentro de los rangos:

- (-Inf,3]
- (3,7]
- (7,Inf]

Así se tendrán tres rangos que describen el nivel de satisfacción, se agregará una columna más con las etiquetas pertenecientes a esta clasificación:

```
> biare_cb$labelP1[biare_cb$catP1 == '(7, Inf]'] <- 'satisfecho'
> biare_cb$labelP1[biare_cb$catP1 == '(3,7]'] <- 'medianamente satisfecho'
> biare_cb$labelP1[biare_cb$catP1 == '(-Inf,3]'] <- 'poco satisfecho'
```

Lo que se pretende con estas tres funciones es la generación de una variable nueva "labelP1" si es que no existe dentro del data frame. Los corchetes limitan a tomar solo las filas que cumplan con un criterio lógico; en este caso, en lenguaje natural:

Toma solo las filas que tengan como valor en la variable catP1 el valor de "'(7, Inf]'" y asigna dentro esta variable y esas columnas el valor de "'satisfecho'".

Así con los otros dos rangos restantes. Una vez que se asignen los tres valores se sabrá qué valores únicos tiene una variable dentro de un data frame usando:

```
> unique(biare_cb$labelP1)
```

- unique(x): Toma como variable un vector, data frame, o arreglo y regresa los valores únicos que tenga este.

Se recibirá como salida las tres etiquetas definidas.

```
> unique(biare_cb$labelP1)
[1] "satisfecho"                "medianamente satisfecho"
[3] "poco satisfecho"
```

Del lado izquierdo, se observa una columna con números. No todas las etiquetas tienen número debido a que es una columna de pistas visuales que refieren a los índices de cada elemento; en este caso, a la etiqueta "medianamente satisfecho" le pertenece el índice número 2.

Obteniendo valores significativos

Ahora con estas categorías y manejo de datos, se puede generar información importante con la librería que se instaló "doBy" de la siguiente manera:

```
> summaryBy(p1 ~ labelP1, data = biare_cb, FUN = descP1)
```

- summaryBy(formula, data, Fun): Esta función calcula de manera grupal un resumen estadístico mediante una fórmula de agrupación y una función la cual aplicar.
    - formula: En este caso el operador de tilde '~' agrupa los elementos de A por cada elemento de B (A ~ B).
    - data: Recibe un data frame como contexto.
    - FUN: Una función estadística compuesta por funciones de agregación que recibirán como argumento cada uno de los grupos que se crearon en la fórmula.

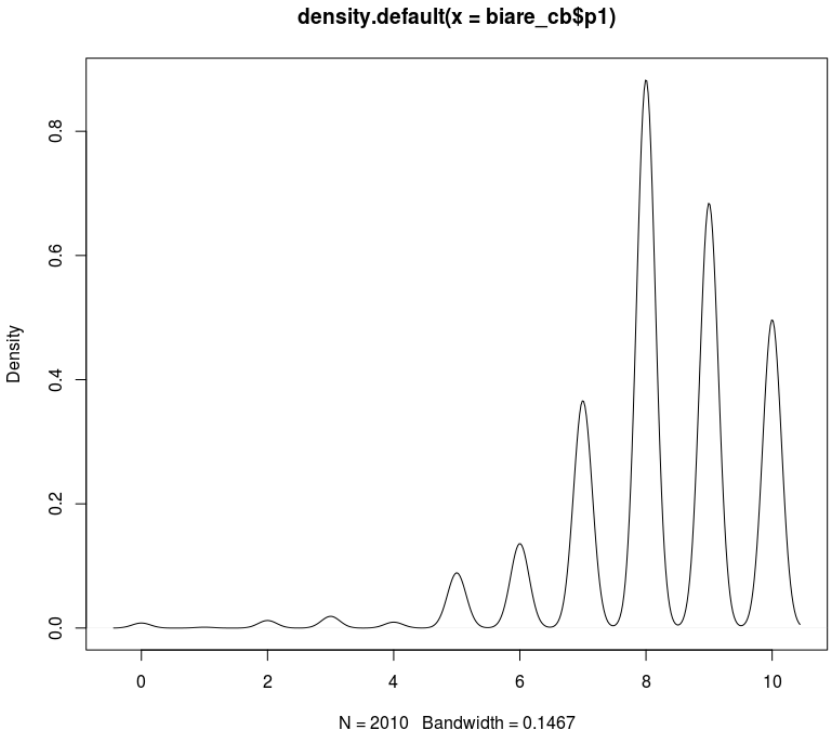
```
> summaryBy(p1 ~ labelP1, data = biare_cb, FUN = descP1)
                  labelP1 p1.FUN1 p1.FUN2  p1.FUN3 p1.FUN4
1 medianamente satisfecho     446       4 6.430493       7
2          poco satisfecho      30       0 2.033333       3
3               satisfecho    1534       8 8.812256      10
```

Al ejecutar la función, se regresa la función descP1 para cada uno de los valores de las categorías. Se puede observar como:

- P1.FUN1 es la cantidad de resultados que se encuentran en dichos rangos.
- P1.FUN2 es el valor mínimo.
- P1.FUN3 es la media.
- P1.FUN4 es el valor mayor.

Estos valores se calcularon para cada una de las tres categorías, representan la cantidad de personas que están poco satisfechas, medianamente satisfechas y muy satisfechas con su vida, respectivamente, acorde a la encuesta. Podemos ver que la mayoría de los mexicanos se consideran satisfechos con su vida.

```
> plot(density(biare_cb$p1))
```



A través de R se puede generar una gráfica de densidad. Esta línea de código toma la variable y usa las funciones:

- density(x, ...): La cual computa la densidad de la función de densidad de probabilidad.
- plot(x): Función con polimórfica, toma un objeto de R y lo gráfica de manera dinámica dependiendo del objeto que se le introduzca como parámetro.

Se toma esta introducción de manera más profunda en la siguiente práctica.

Conclusión

En esta práctica introductoria se cubrió qué es el análisis de datos y por qué es importante tener herramientas computacionales para generar información a partir de datos que carguemos con el ambiente del lenguaje R; a su vez, se evidenció su utilización en la industria e instalación en diferentes plataformas.

También se introdujeron los elementos básicos de R: variables, data frames, operadores y funciones.

Una alternativa de práctica propuesta sería cargar una base de datos, tenerla en memoria como un objeto dentro del ambiente y poderla ir manipulando y estar monitoreando los cambios y objetos en la interfaz gráfica.

Además, se declaró la función propia con sub funciones y se crearon subconjuntos mediante reglas de clasificación básica de los datos iniciales para etiquetar las clasificaciones de satisfacción.

Se generó información a partir de los datos, para esto se exploraron; se hizo una deducción y clasificación de una variable para inferir suposiciones básicas de la misma.

Ejercicio

Al igual que se hizo con la función estadística anterior, se necesita saber cuál es la media, mediana y desviación estándar sobre la pregunta número uno y dos (“Y hace un año, ¿qué tan satisfecho se encontraba con su vida?”).

Con base en estas funciones concluir: ¿Crees que ha mejorado la percepción de los mexicanos con su vida?

Práctica 2. Gráficas con ggplot2

Introducción

Es importante para el **análisis de datos** que su representación sea concreta y significativa; por lo tanto, se busca transformar los datos a una gráfica con el objetivo de resaltar información útil, facilitar el análisis concluyente y, con ello, la toma de decisiones.

Para construir una gráfica, se requiere considerar la estructura de los datos y suponer una relación de estos. Aun cuando dicha inferencia fuese incorrecta nos dará indicios para identificar correspondencia entre ellos.

En esta práctica se usa la paquetería ggplot2. Paquetería robusta creada por Hadley Wickham basada en la gramática de gráficas de Leland Wilkinson, la cual genera gráficas complejas por capas.[11]

Los componentes clave de ggplot2 incluyen:

- **Data frame (marco de datos)**: Es la fuente de datos para crear gráficos en ggplot2. El marco de datos debe contener las variables que se utilizarán en el gráfico.
- **Geometrías (geoms)**: Representan la forma en que los datos se visualizan en un gráfico. Pueden ser puntos, líneas, barras, áreas, entre otros. Por ejemplo, *geom_point()* se utiliza para crear un gráfico de dispersión, *geom_line()* para un gráfico de líneas y *geom_bar()* para un gráfico de barras.
- **Estéticas (aesthetics)**: Son las propiedades visuales de los elementos en el gráfico, como el color, la forma, el tamaño o el grosor. Las estéticas se definen dentro de la función *aes()* y se asignan a variables del marco de datos. Por ejemplo, *aes(x = variable_x, y = variable_y)* asigna las variables "variable_x" y "variable_y" al eje x e y, respectivamente.

[11] https://ggplot2.tidyverse.org/

- **Escalas (scales)**: Controlan cómo se mapean los valores de los datos a las propiedades visuales, como el rango de ejes, colores o tamaños. ggplot2 ajusta automáticamente las escalas según los datos, pero también es posible personalizarlas utilizando funciones como *scale_x_continuous()* o *scale_fill_manual().*
- **Facetas (facets)**: Permiten dividir un gráfico en paneles según los valores de una o varias variables. Esto es útil cuando se desea comparar diferentes grupos o subconjuntos de datos en un solo gráfico. Se pueden agregar facetas mediante la función *facet_wrap()* o *facet_grid()*.
- **Temas (themes)**: Controlan el aspecto visual general del gráfico, como los colores, las fuentes, los tamaños de texto y las líneas de cuadrícula. ggplot2 ofrece varios temas predefinidos, como *theme_gray()* o *theme_minimal()*, pero también se pueden personalizar utilizando la función *theme()*.

Estos son algunos de los componentes principales de ggplot2 en R. Al combinarlos de manera adecuada, se puede crear una amplia variedad de gráficos y visualizaciones efectivas para explorar y comunicar sobre los datos.

Objetivo

Que el alumno se familiarice con la paquetería y la gramática de las gráficas.

Una vez conocidos los componentes de ggplot2, se instalará la paquetería y empleará para la creación de gráficos descriptivos a los que se agregarán capas. Además, se continuarán utilizando los datos en memoria para dar seguimiento al flujo de conocimiento generado a partir de esta base de datos del BIARE.

Requerimientos

1. **Instalación de ggplot 2**

Se instala la paquetería ggplot de la misma manera que se realizó en la práctica uno.

```
> install.packages("ggplot2")
```

El ambiente de desarrollo de R se encargará de descargar e instalar todas las dependencias que esta paquetería necesite. Una vez terminado el proceso, se cargará la librería.

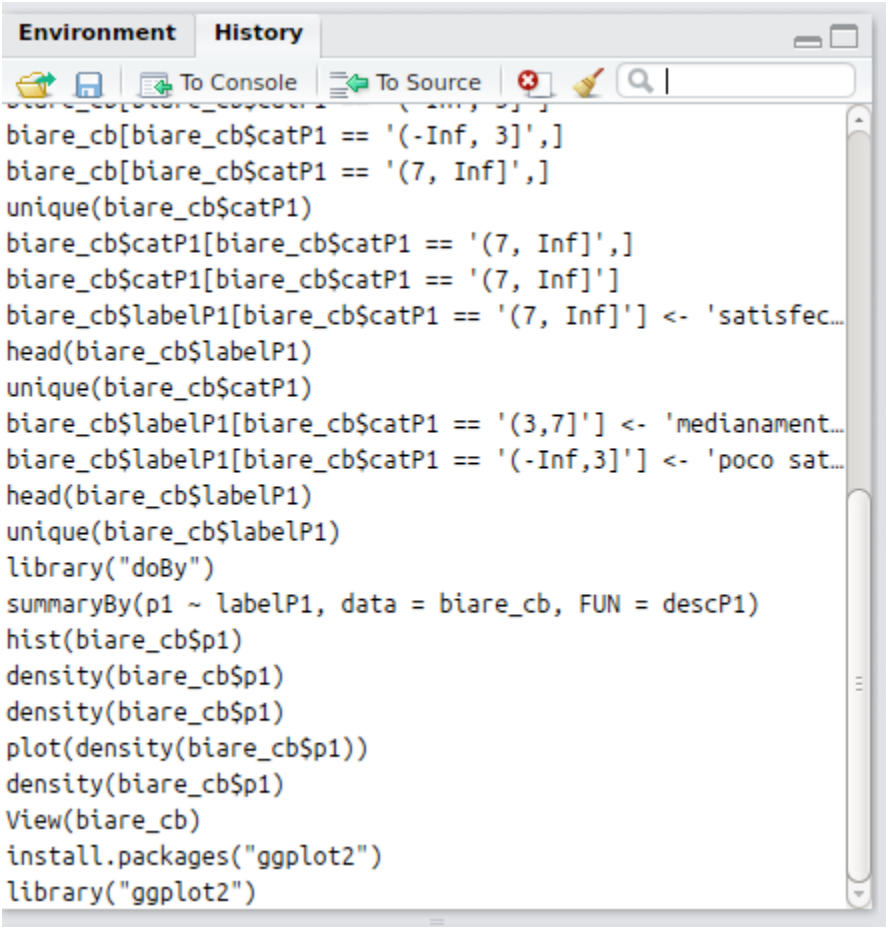
```
> library("ggplot2")
```

Si después de la carga no se recibe respuesta, será un indicativo de su correcta instalación y carga.

2. Carga de los datos iniciales

Primero, se revisará que los datos a utilizar estén cargados en la memoria que se utilizará.

Se dará seguimiento a la práctica pasada, con el empleo de la misma base de datos incluyendo las modificaciones que se realizaron. En caso de requerirlo, dichas modificaciones se pueden consultar en la pestaña del historial de la consola que se encuentra en la ventana donde se consultaron previamente los objetos en memoria (environment).



Dessarollo

1. Preparación de los datos

Como se usarán datos personales de los encuestados y estos se encuentran en otro archivo CSV, también se importará a R como se realizó anteriormente en un nuevo data frame al cual se asignará el nombre de “biare_cs”.

```
> biare_cs <- read.csv("/home/user/Downloads/biare_cs_0717.csv")
```

De esta manera, se tendrán dos bases de datos cargadas. Estas bases están relacionadas por una llave, como explica en el pdf de la estructura de base de datos.

En la llave, ambas bases de datos están conformadas por las columnas:

- FOL
- ENT
- CON
- V_SEL
- N_HOG
- H_MUD
- N_REN

Entonces, para tener un nuevo data frame que contenga ambas bases de datos mezcladas y relacionadas por esta llave se usará la función *merge*.

- merge(x, y, by = c("x.1","y.1")): Recibe como parámetro dos data frames y las combina por una o varias columnas. En bases de datos relacionales se le conoce como *join.*

Con esta función se generará un nuevo data frame que contendrá esta unión:

```
> biare <- merge(biare_cs, biare_cb, by =
c("fol","ent","con","v_sel","n_hog","h_mud","n_ren"))
```

Se unirán los dos data frames en uno nuevo "biare", que contendrá las filas del data frame con menos elementos, sería como hacer un *inner join* en una base de datos relacional. Se puede consultar la estructura de nuestro data frame en la ventana de elementos.

2. Gráficos sin ggplot

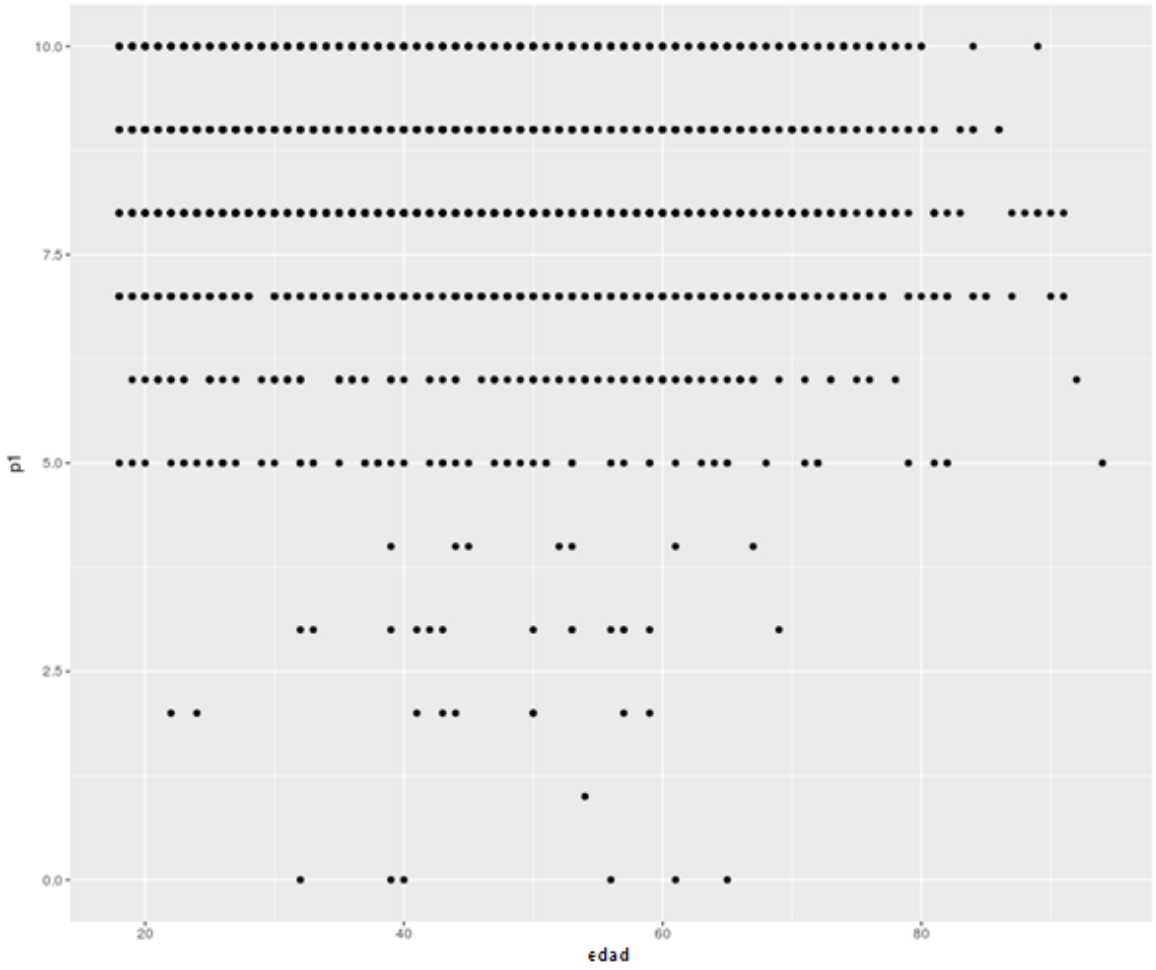
Una vez que se tenga la certeza de que la estructura del nuevo data frame es correcta, se escribirá la primera instrucción para la creación de la gráfica inicial en ggplot.

```
> qplot(edad, p1, data = biare)
```

Lo que se hizo fue usar la función qplot.

- qplot(x, y, data=): Recibe dos variables conjunto a sus data frame de origen y las grafica de manera rápida y dinámica.

Con los argumentos edad y la respuesta a la pregunta 1 se evaluará la existencia de una relación o tendencia entre ellas. Se tendrá como resultado una gráfica parecida a la siguiente:

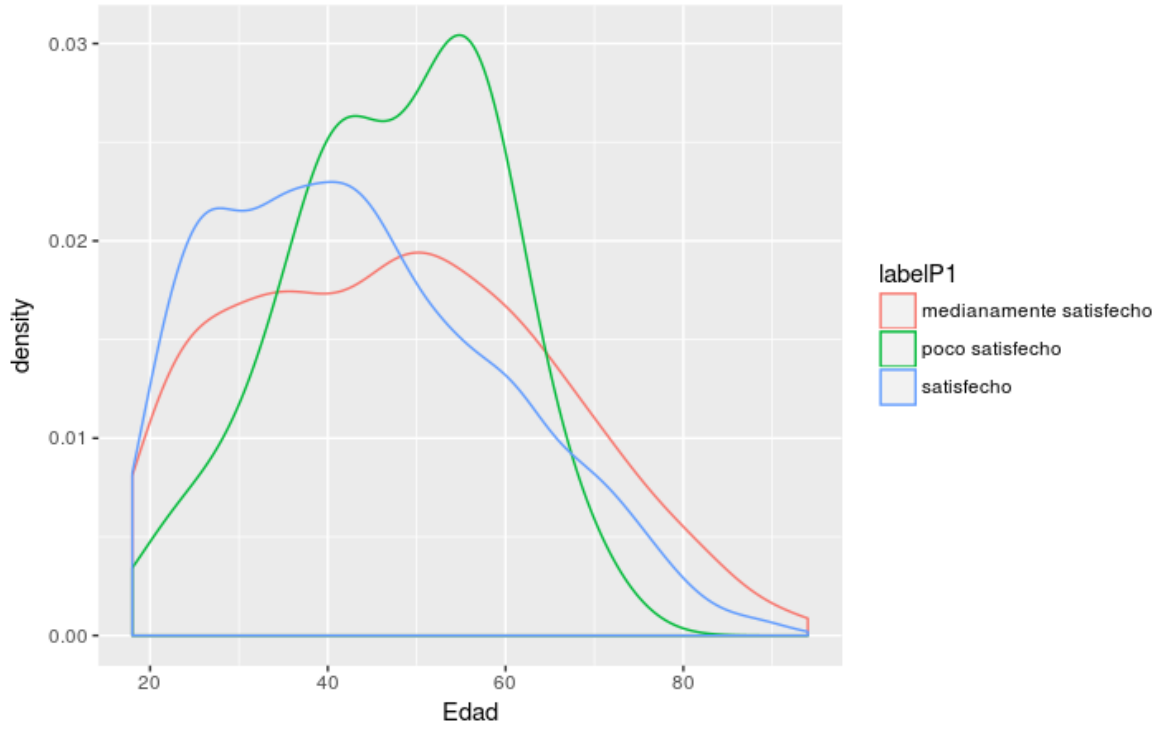


Lo que se aprecia es que antes de los 30 y después de los 70 casi nadie se considera poco satisfecho con su vida.
Es posible personalizar esta función de manera limitada; sin embargo, de manera directa resulta muy útil para hacer gráficos rápidos.

Si se escribe lo siguiente se puede tener una mirada más profunda al comportamiento de las categorías que se dieron a través de la edad de los encuestados.

```
> qplot(edad, data = biare, geom = "density", colour = labelP1, xlab="Edad")
```

Aquí se especifican como parámetros que será una gráfica de densidad de población, que se nombrará al eje de las x como “Edad” y que se dividirán en grupos a cada una de las categorías de satisfacción que se tengan. Así, el programa regresará una gráfica parecida a esta:



3. Gráficas por capas con ggplot

Ahora, se procederá a construir una gráfica similar por capas con la función principal de ggplot2, la cual es precisamente “ggplot”.

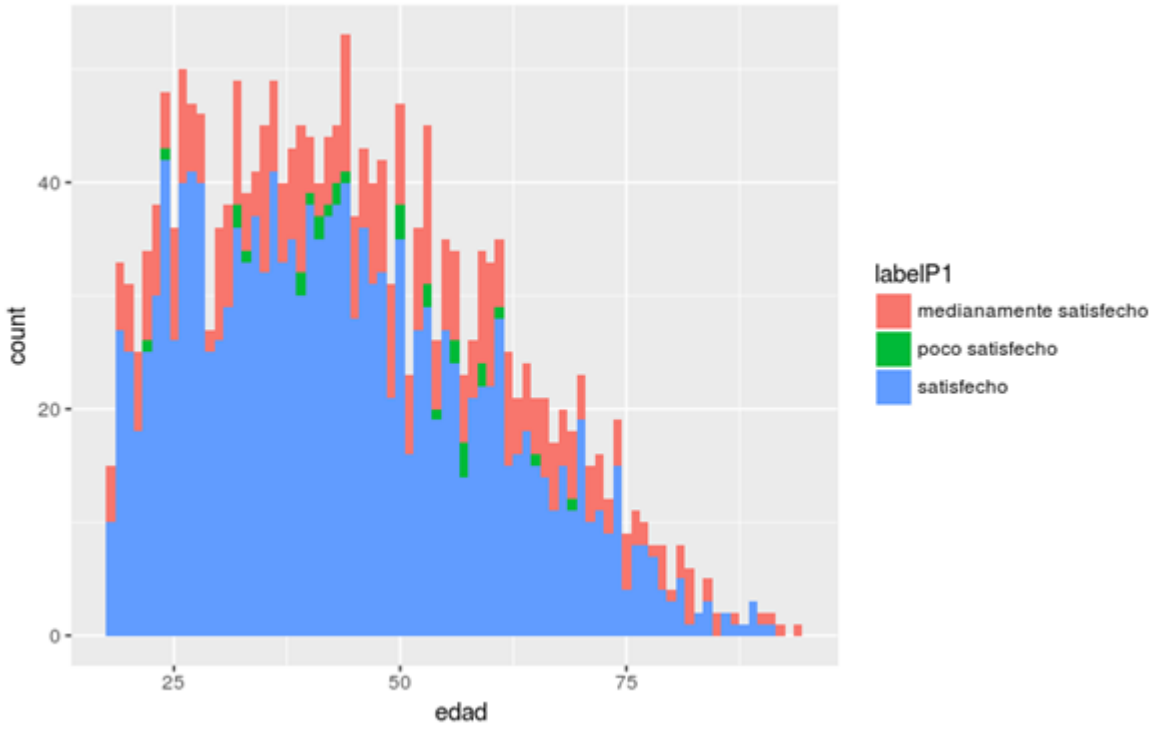
```
> ggplot(biare, aes(x=edad, fill=labelP1)) + geom_histogram(binwidth=1)
```

Aquí se observa el primer ejemplo de gráfica por capas.

Todas las gráficas que usen ggplot() llevan los datos que les hayan sido asignados; en esta, especificaremos qué datos se emplearán para graficar y la función aes() que provee a la gramática el mapeo de los elementos estéticos que llevará la gráfica a generar.

Una vez que la estructura de la gráfica tenga base, se podrán agregar las capas. En este caso, se agregará una capa geométrica de histograma y como parámetro le se pasará el ancho de las barras.

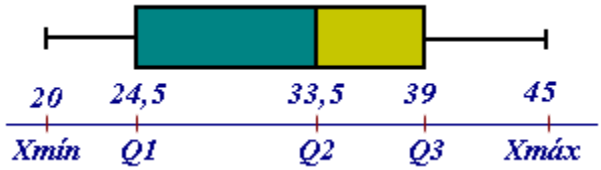
La gráfica se vería así.



Una excelente manera de visualizar distribuciones por grupos es el empleo de gráficas de cajas y bigotes, pues describen muy bien el comportamiento de los grupos y categorías que se van generando.

Este tipo de gráficas también se pueden generar mediante una capa de geometría distinta.

4. Gráfica de cajas y bigotes con ggplot



Teniendo en cuenta Q1, Q2, Q3 como el primer, segundo y tercer cuartil de una distribución normal. De Xmin a Q1 es Cuartil 1, de Q1 a Q2 es el segundo Cuartil y así los siguientes dos. El primer y último cuartil, según el modelo de distribución normal, suelen ser los cuartiles con mayor dispersión; es decir, que su desviación estándar es más grande y por lo tanto están más alejados de su promedio (Estadística para todos, 2017).

Xmin y Xmax como los valores mínimo y máximo respectivamente.

El primer cuartil divide el 25% más pequeño del resto de los datos. El segundo cuartil divide el 50% más pequeño del resto de los datos, también llamada la media. El tercer cuartil divide el 75% más pequeño del resto de los datos.

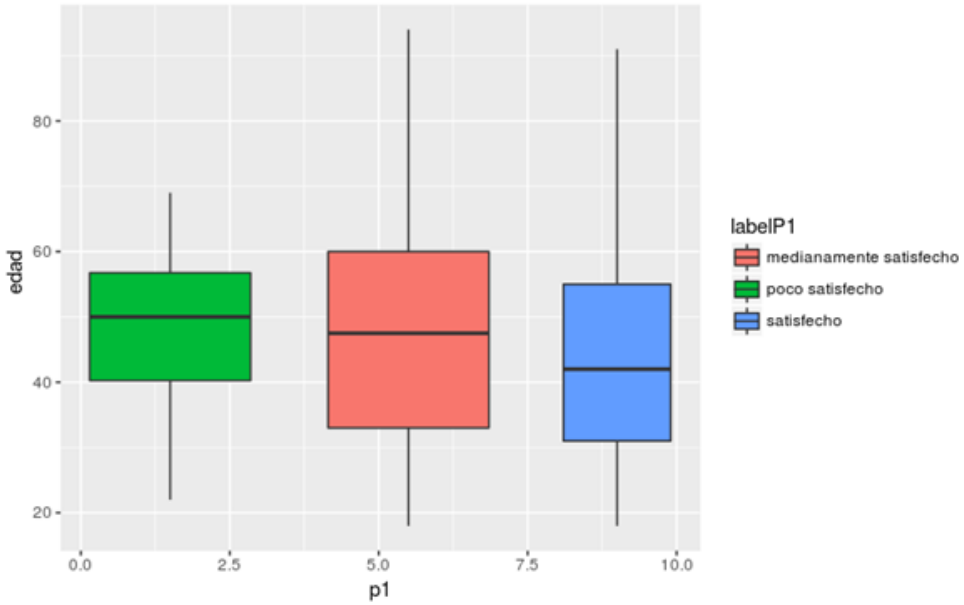
El bigote de la izquierda representa al colectivo de datos (Xmín, Q1) menor al primer cuartil.

El bigote de la derecha representa al colectivo de datos mayores al tercer cuartil.

Este tipo de gráficas sirve para observar diferentes poblaciones de datos en la misma gráfica con fines comparativos.

Así se podrá generar una gráfica por categorías de satisfacción previamente creadas.

```
> ggplot(biare, aes(x=p1, y=edad, fill=labelP1)) + geom_boxplot()
```



Así se puede observar eficientemente la distribución de las respuestas de los encuestados, cómo las respuestas positivas tienden a ser más altas y cómo la mayoría de las personas que están poco satisfechos con su vida se encuentran entre 40 y 50 años de edad.

Este tipo de gráficas son descriptivas. R no sólo sirve para explorar datos y observarlos, también es útil para hacer predicciones e intentar modelar comportamientos.

Conclusión

Al final de la práctica se observó cómo una serie de datos, una vez limpios y lógicamente relacionados, nos pueden ofrecer una gran cantidad de información sobre el giro del negocio o investigación de donde provienen los datos.

Resumiendo, en esta práctica se enseñó como:

- Unir un par de data frames.
- Instalar la paquetería ggplot2.
- Graficar datos incluyendo gráficas rápidas y gráficas por capas.
- Interpretar gráficas para generar información en favor del giro del negocio o investigación.

Con estas habilidades básicas se espera que el alumno indague más en la interpretación de datos y use las funciones de manipulación para generar estadísticos de alto valor representativo.

Ejercicio

Se quiere responder cómo afecta la cantidad de años de educación básica terminados con la percepción de satisfacción y, también, la influencia de la identidad de género con esta satisfacción.

Represente de la mejor manera posible construyendo gráficas que describen estas relaciones, use geometrías que no se hayan usado y concluya si existe realmente alguna relación.

Práctica 3. Análisis exploratorio de datos espaciales con R

Introducción

Se generará un mapa temático capaz de representar gráficamente información seccionada por la división política de los estados dentro de la República Mexicana.

Esta información será demográfica y geográfica.

La base de datos que se usará fue descargada de los servidores del INEGI. Se manipularán con librerías del lenguaje R los datos para generar información concreta y concisa sobre la nación.

Requerimientos

Para esta práctica se necesitarán las librerías siguientes: (Anexo 1)

1. *ggmap*: Extiende las capacidades de ggplot2 para graficar mapas.
2. *rgdal*: Una interfaz a la popular librería de proceso de datos espaciales de C/C++ gdal.
3. *rgeos*: Una interfaz a la poderosa librería de procesamiento vectorial geos.
4. *maptools*: Provee funciones de mapeo.
5. *dplyr*: Una herramienta rápida y consistente para trabajar con datos como objetos, tanto en memoria como fuera de memoria.
6. *tidyr*: Una herramienta para la incorporación de datos en variables columnares, funciona bien con 'Dplyr' tuberías de datos.
7. *tmap*: Nueva paquetería para crear mapas magníficos.

Como se realizó en la práctica anterior para carga cualquier paquetería se usará la función **library()**, si la función no entregase cadena de texto sería indicio de que se importaron las librerías necesarias.

En caso de que el proceso devolviera una cadena de error, muy probablemente indique que no han sido instaladas las librerías. Se tendría que usar la función **install.packages()** la cual recibirá como argumento la librería que se tiene que instalar antes de importar.

Atajo: hay una manera rápida de instalación e importación para todas las librerías descritas con sólo un vector de la siguiente manera:

a) **x <- c("ggmap", "rgdal", "rgeos", "maptools", "dplyr", "tidyr", "tmap")**
b) **install.packages(**x**)** # esto se podría demorar algunos minutos, depende de la conexión a Internet.
c) **lapply(x, library, character.only = TRUE)** # se importarán las librerías necesarias.

Desarrollo

1. Datos espaciales en R

Ahora, se conocerá y explorará el formato empleado por las bases de datos espaciales para extraer, transformar, cargar (ETL) y graficar los datos.

De inicio, se conseguirán estas bases de datos.

En la información espacial Internet tiene muchos recursos gratuitos, libres y bajo licencias de software libre. Uno de los formatos más populares para bases de datos espaciales es “shapefiles”, realmente es un formato de archivo geométrico.

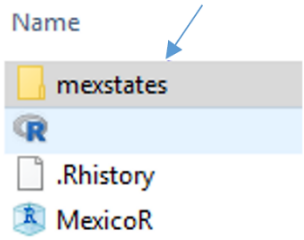
Dentro del interpretador de R hay muchas maneras de importar este tipo de archivos, la manera más popular es con la función **readOGR** de la librería **rgdal.** De la misma manera, podría ser importado un archivo .shp con la función **readShapePoints()** de la paquetería **maptools.**

rgdal es la interfaz de R a la “Geospatial Abstraction Library (GDAL)” usada por software open source del tipo GIS como QGIS (Quantum GIS). Soporta a R en el manejo de un amplio rango de formatos de archivos.

Se descargará un archivo shapefile del siguiente enlace:

https://www.inegi.org.mx/app/biblioteca/ficha.html?upc=889463770541

Se descomprimirá en la carpeta de un nuevo proyecto de RStudio de la siguiente manera:



Se colocará en el proyecto creado, en este caso R_ESPACIALES.

Se revisa el formato de los archivos que tenga la base de datos espacial y verifica que tenga compatibilidad con readOGR.

Para importar los datos en RStudio se escribe la función de la siguiente manera:

Ind <- readOGR(dsn = "mexstates", layer = "mexstates")

En el primer argumento de la función dsn (data source name) se pondrá la ruta de donde se encuentra el archivo espacial; luego, layer al cual se le pasará el nombre del archivo de la base de datos espacial. No es necesario poner el formato del archivo, esta función identificará la mejor manera de importar los archivos.

En R no se requiere definir el nombre del argumento antes de pasárselo, aunque se recomienda como buena práctica definir estos nombres para la legibilidad de las funciones.

2. La estructura de datos espaciales en R

Una vez creado el objeto espacial lnd se puede ver que está conformado de diferentes ranuras (slots) siendo los más importantes @data y @polygons . La ranura @data se ve como una tabla normal en una base de datos, con filas y columnas. Y la ranura @geometry es donde se almacenan los vértices de los polígonos.

Veamos cómo está conformada la ranura de @data:

head(lnd@data, n = 2)

Con el argumento n se traerán las dos primeras filas de la ranura @data, se observa cómo están conformados los datos.

3. Gráfica básica

Ahora que se conoce la estructura de datos clásica de la ranura de datos, sigue graficar el mapa; se hará con una de las funciones más poderosas y versátiles de R

plot(lnd)

Se pasó como argumento el objeto espacial y automáticamente la función plot sabrá graficarlo y nos entregará una serie de polígonos vectorizados como estos:

Como se puede apreciar en la imagen, plot es una función inteligente que construye las gráficas requeridas; además, es personalizable.

4. Selecciones básicas y espaciales

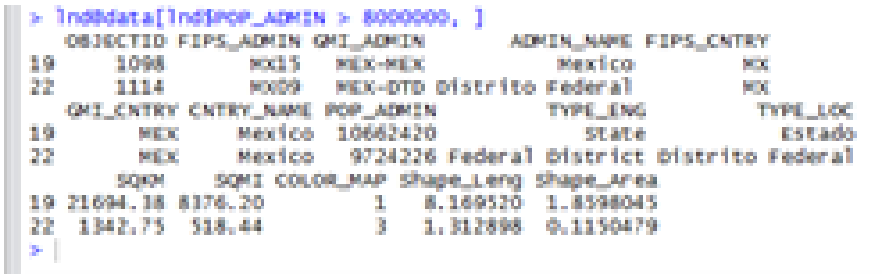
El siguiente paso será hacer una selección de los datos con un criterio fijo.

Para la base de datos que se importó, se seleccionaron los estados con una población mayor a los ochocientos mil habitantes del atributo de población:

Ind@data[Ind$POP_ADMIN > 800000,]

En esa línea de código lo que se hace es una consulta a la ranura del objeto espacial usando el operador de selección [], el cual recibe dos cosas: el criterio de selección y, después de la coma, los atributos (columnas que se deseen traer) ya sea por su posición en la ranura o por su nombre.

R, al ser un lenguaje funcional, permite guardar sentencias dentro de variables para luego ejecutarlas dentro de “subsets”, por ejemplo:



```
> Ind@data[Ind$POP_ADMIN > 8000000, ]
   OBJECTID FIPS_ADMIN GMI_ADMIN       ADMIN_NAME FIPS_CNTRY
19     1098       MX15   MEX-MEX           Mexico         MX
22     1114       MX09   MEX-DTD Distrito Federal         MX
   GMI_CNTRY CNTRY_NAME POP_ADMIN         TYPE_ENG          TYPE_LOC
19       MEX     Mexico  10662420            State            Estado
22       MEX     Mexico   9724226 Federal District Distrito Federal
       SQKM    SQMI COLOR_MAP Shape_Leng Shape_Area
19 21694.38 8376.20         1   8.169520  1.8598045
22  1342.75  518.44         3   1.312898  0.1150479
>
```

Sel <- Ind$POP_ADMIN < 8000000 & Ind$POP_ADMIN > 3000000

Ind@data[Sel , c(4,8,11)]

ADMIN_NAME POP_ADMIN SQKM
2 Nuevo Leon 3370912 65173.05
8 Jalisco 5772704 79851.44
10 Veracruz 6774736 71286.36
11 Guanajuato 4332525 30466.18
17 Puebla 4487672 34365.46
18 Michoacan 3859507 59617.63
25 Oaxaca 3329574 92715.81
27 Chiapas 3524501 73771.48

Se observa claramente la posibilidad de unir sentencias en su homónimo en SQL AND con el operador &. Están declaradas en la variable “sel”, la cual se pasará al operador junto con el vector que contendrá la dimensión de nuestra tabla como segundo argumento y las columnas de la ranura de datos que se apetezca extraer.

5. Funciones espaciales

Al igual que varias extensiones espaciales, rgeos provee de funciones aplicables a la minería de objetos espaciales. Ahora, se determinará el centroide geométrico de México:

gCentroid(Ind)

```
        x       y
1 -102.5329 23.95046
```

Se entregarán las coordenadas geométricas del centroide, para su visualización es importante estructurarlos dentro de plot.

```
> gCentroid(Ind)
class       : SpatialPoints
features    : 1
extent      : -102.5329, -102.5329, 23.95046, 23.95046  (xmin, xmax, y
min, ymax)
coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs8
4=0,0,0
>
```

Sin embargo, si se sigue el argumento siguiente:

plot(gCentroid(Ind))

Se generará un punto solamente, donde se encuentra el centroide en nuestras coordenadas relativas.

Entonces, primero se tendrá que dibujar la serie de polígonos que forman a México:

plot(Ind)

Después, se agregará el centroide con la función points(). La cual permite agregar puntos con coordenadas fijas a la gráfica que se tenga previamente desplegada.

Se hace de la siguiente manera:

points(gCentroid(Ind), col = “blue”)

Para que, con esas dos líneas se otorgue un mapa así:

En azul, se aprecia el centroide geométrico del mapa. Lo que confiere la posibilidad de agregar puntos de diferentes colores con el parámetro ‘col’, parámetro que también existe en plot y lines.

Al paradigma que es la programación funcional se puede seguir apilando capas de puntos de otros colores que se asignen como argumento y donde sea requerido.

Para saber qué argumentos puede recibir la función o que hace la función misma se consultará en la consola de R la documentación, anteponiendo un signo de interrogación cerrado antes del nombre la función, de esta manera:

?points

Si no se tiene instalada la paquetería, se recomienda consultar en Internet en los repositorios oficiales de R sobre la misma función anteponiendo dos signos de interrogación cerrada, de la siguiente manera:

??points

6. Consultas espaciales complejas

Ahora:

- ¿Y sí queremos el centroide de cada estado?
- ¿Qué tal aquellos estados cuyo centroide esté a menos de 600 kilómetros del centroide de la ciudad de México?

De inicio, se conseguirá consultar sólo el centroide de la ciudad de México. Se seleccionará la fila que se asocie al polígono buscado.

Se usará la función llamada 'grep', funciona muy parecida al programa de UNIX, de la siguiente manera:

Ind@data[grep("Distrito", Ind$ADMIN_NAME),]

df <- Ind@data[grep("Distrito", Ind$ADMIN_NAME),]

Como ya se vio antes, lo que se hace es seleccionar de la ranura de datos todas las filas que cumplan con una sentencia de selección. En este caso, la sentencia es una función, la función grep(), que toma como argumento una cadena de caracteres y un campo de un dataframe.

En este caso, grep decuelve el número identificador de la fila o filas que tengan en su campo de nombre administrativo el segmento de caracteres "Distrito".

grep() es tan poderoso que incluso recibe regex (anexo) como argumento de cadena. Eso va más allá del alcance de esta práctica; de momento, se limitará a desplegar la fila de distrito federal.

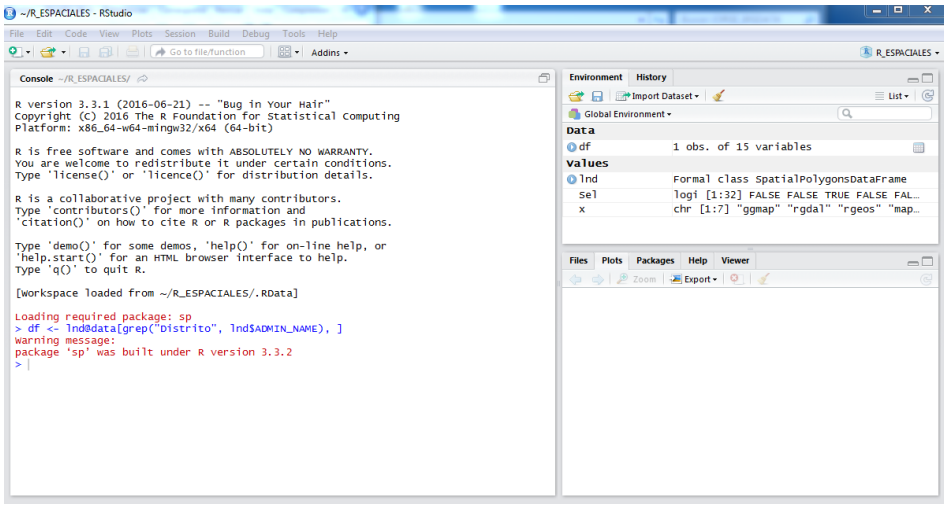
```
> Ind@data[grep("Distrito", Ind$ADMIN_NAME), ]
   OBJECTID FIPS_ADMIN GMI_ADMIN       ADMIN_NAME FIPS_CNTRY GMI_CNTRY
22     1114       MX09   MEX-DTD Distrito Federal         MX       MEX
   CNTRY_NAME POP_ADMIN         TYPE_ENG         TYPE_LOC    SQKM   SQMI
22     Mexico   9724226 Federal District Distrito Federal 1342.75 518.44
   COLOR_MAP Shape_Leng Shape_Area
22         3   1.312898  0.1150479
> |
```

Se guardará esta sentencia en una variable para facilitar su manejo.

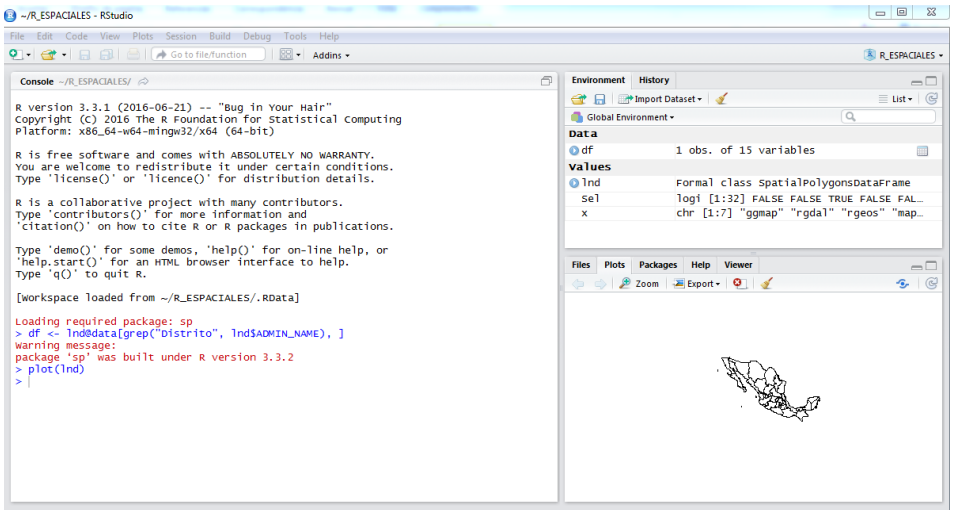
df <- Ind[grep("Distrito", Ind$ADMIN_NAME),]

Nótese como se guardó la sentencia, pues no fue exactamente igual que su visualización. En este caso, no sólo se seleccionó la ranura de datos (Ind@data) sino también todo el objeto espacial para que se mantenga la relación con sus polígonos.

Perfecto, ahora se podrá seleccionar de la fila destacada su centroide para comparar con las demás. Se probará graficarlo:



plot(Ind)



points(gCentroid(df), col= "yellow")

Se generará el mapa de México con el centroide de CDMX resaltado en amarillo.

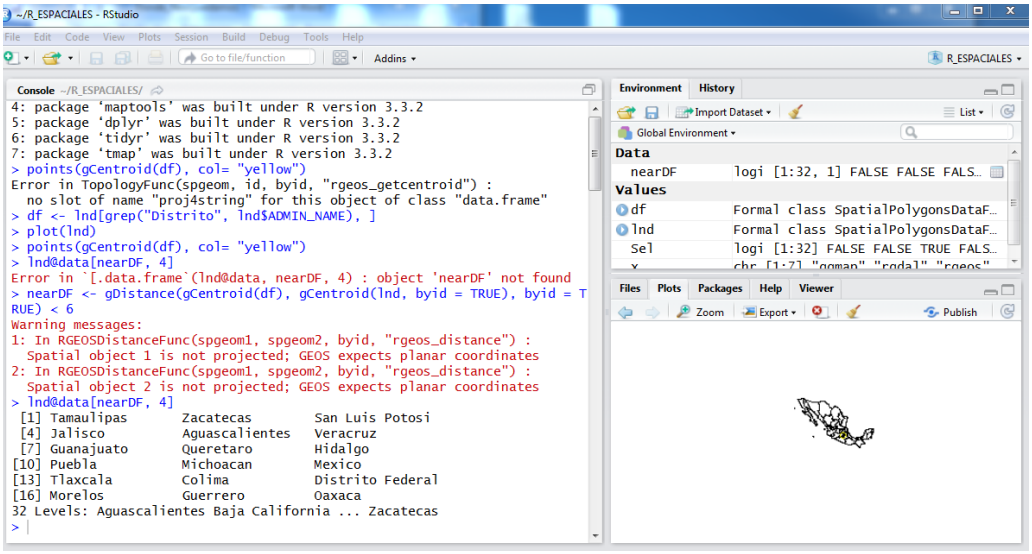
El siguiente paso, es definir la sentencia de selección por la cual se elegirán los estados que tengan la distancia buscada. Se hará con la función gDistance(), que acepta como argumento principal dos geometrías y devuelve la distancia entre ellas:

nearDF <- gDistance(gCentroid(df), gCentroid(Ind, byid = TRUE), byid = TRUE) < 6

Ambas funciones tienen como argumento también 'byid', la cual recibe un valor booleano y determina si debe aplicar la función a cada una de las sub geometrías (TRUE) o a la geometría padre solamente (FALSE).

No hay que intimidarse por la complejidad del uso de paréntesis, que pareciera excesivo. Los lenguajes funcionales siempre emplearán cuantiosos paréntesis, lo que se pasa a la variable nearDF es una sentencia de selección: que seleccione filas cuya distancia entre el centroide del DF y el centroide de cada estado sea menor a seis km.

La bandera “byid” que seguramente alinearía a cualquiera, lo único que hace es realizar la misma función. Solo que, en lugar de hacerlo en la figura general lo ejecuta por cada polígono del cual esté compuesta la geometría grande (por cada estado en este caso).



NOTA: Aun marcando warning, No hay problema

Se observa qué estados son esos mediante una sub selección del dataframe, mediante corchetes, ponemos una función lógica de filtrado y, después de la coma, el vector de columnas que se desee seleccionar:

lnd@data[nearDF, 4]

Se tendrán de regreso los estados cuyo centroide está a seis o menos kilómetros del CDMX (la primer columna de números entre corchetes sólo son índices para llevar cuenta, línea por línea, en qué índice se encuentra).

[1] Tamaulipas Zacatecas San Luis Potosi Jalisco
[5] Aguascalientes Veracruz Guanajuato Queretaro
[9] Hidalgo Puebla Michoacan Mexico
[13] Tlaxcala Colima Distrito Federal Morelos
[17] Guerrero Oaxaca

Sólo se trasladó la columna cuatro, que corresponde al nombre del estado.

¡Perfecto, el resultado resulta coherente! Ahora, faltaría su graficación.

Se comenzará con la creación de un objeto espacial sólo con los estados que cumplan la sentencia requerida de la siguiente manera:

nearStates <- Ind[c(nearDF), 4]

Nótese cómo será transformada la matriz de selección en un vector con la función c() dado que, al seleccionar un objeto espacial no es posible pasar una matriz como argumento, tiene que ser un vector lógico.

Se necesita graficar el país entero, para esto se emplea la función:

plot(Ind)

Luego, se graficará sobre ésta los estados que se requiere resaltar:

plot(nearStates, col = "blue", add = TRUE)

Es crucial activar la bandera “add” para que el nuevo gráfico se adhiera a la anterior, sin ello la gráfica entera (el país completo) se borraría.

Así se ven los estados cuyo centroide está a menos de seiscientos kilómetros del centroide de CDMX.

Conclusión

La visualización de datos geográficos ayuda en la toma de decisiones, desde localización de comercios o mejores rutas para movernos en la ciudad en aplicaciones de mapas en nuestro dispositivo móvil, hasta la descripción detallada de sismos.

Lo poderoso de todas las herramientas de minería de este tipo de datos son las relaciones que podemos observar entre los diferentes tipos de datos que contienen nuestras bases de datos geográficas. Además, la posibilidad de localizar funciones de agregación directamente en el mapa nos permite tener un mejor entendimiento de fenómenos que pueden no verse en primera instancia y este tipo de herramientas los hacen más evidentes y lógicos. Una de las

aplicaciones es en epidemiología, donde se quiere localizar y aislar núcleos de infección.

Por ello, la conveniencia de que los Ingenieros de Datos cuenten con la habilidad de transformarlos en interfaces que resulten más amables para apoyar a los profesionales no especializados en estas ramas en la toma de decisiones.

Ejercicio

Una de las tareas más comunes del análisis espacial es encontrar intersecciones entre polígonos.

En la presente práctica estos polígonos son estados; por lo tanto, se desea encontrar qué estados colindan con el estado que cuenta con mayor población.

Graficar y observar cuál es la población de cada uno de los estados colindantes y ver si existe alguna relación entre dichas poblaciones.

Práctica 4. Virtualización de datos

Introducción

Hasta donde se sabe, existe una relación directa entre la temperatura y el volumen de distintos materiales, estos se dilatan al calentarlos y se contraen al enfriarlos.

Normalmente, se supone que el coeficiente de dilatación térmica es constante; aunque no sea estrictamente cierto, se sabe que existe una relación lineal y esta aproximación para un gran número de aplicaciones es aceptable.

Este acercamiento a la constante de dilatación térmica es determinada por una relación lineal, la cual recibe como variable el cambio en la temperatura y entrega el cambio en longitud del material en estudio. Como ejemplo en este caso se usará una barra de acero.

Dado que cada vez que se modifica la temperatura en nuestro sistema se modifica linealmente la longitud, es viable emplear la ecuación de la recta como base pues resulta en el modelo de una clásica relación lineal entre dos variables:

$$f(x) = mx + b$$

De esta manera, el fenómeno en estudio puede ser descrito según el siguiente modelo matemático:

$\Delta L = (\alpha \cdot \Delta T) + 1$

$Lf = Li\ (1 + \alpha \cdot \Delta T)$

Donde:

α = Coeficiente de dilatación lineal [$°C^{-1}$]

Li = Longitud inicial

Lf = Longitud final

ΔL = Delta de longitud = Longitud final – Longitud inicial

ΔT = Delta de temperatura = Temperatura final – Temperatura inicial

En la experimentación propuesta, se trabajará con la misma varilla por lo que la longitud inicial y coeficiente de dilatación serán constantes y tomarán los valores de 10 m y 1.1 $x10^{-5}$ $°C^{-1}$ respectivamente.

El valor del coeficiente de dilatación para el acero no resulta en un consenso dentro de la diversas fuentes en las que es posible consultar. De manera general, este valor oscila entre 11.0 $x10^{-6}$ $°C^{-1}$ y 12.0 x 10^{-6} $°C^{-1}$; razón por la cual resulta conveniente adaptar los datos virtuales a esta desviación, así se generarán datos más próximos a la realidad.

Requerimientos

Se necesita un conjunto de datos semi aleatorizados que emule un experimento metrológico para la dilatación lineal de una varilla de acero en relación a la variación de la temperatura. Los datos serán generados con una distribución normal y con una determinada desviación estándar.

Creación de vectores

Se emplearán listas de las variables para el manejo en R, ello con la creación de vectores.

El primer vector de valores requerido es el cambio de temperatura o delta de temperatura ΔT.

La escala empleada será Celsius, por lo que se expresarán las temperaturas en grados centígrados (°C), la fluctuación que adquirirá esta variable irá desde 1 °C y hasta 100 °C y sus valores serán generados de manera fija en aumentos discretos de un grado centígrado.

Se realizará de la siguiente manera:

```
> x <- (1:100)
```

El segundo vector de valores corresponde al cambio en la longitud o delta de longitud ΔL, a expresarse en metros (m). Dado que esta es resultante de la variación de la temperatura acorde al modelo de dilatación lineal, se trata de la variable dependiente 'y'.

Para su asignación, se empleará una función de transformación que se identificará para facilitar el seguimiento de la transformación desde dentro de los paréntesis hacia afuera:

```
> y <- 10 * (
        1 + (
                    ( rnorm(100, sd = 0.00000005) +
        0.00000115 ) * x
        ))
```

Ahora, para la obtención de valores aleatorios se utilizará la función 'rnorm'.

Se recuerda que la labor de esta función es tomar como argumentos los datos y una distribución específica.

Como el primer argumento es un número entero 'x', el cual es la longitud del vector, y el segundo es la distribución estándar de la distribución normal. Será entregado un vector de dimensión 'x', cada uno de sus elementos se encuentra en la distribución normal con la desviación estándar definida de 5×10^{-8}.

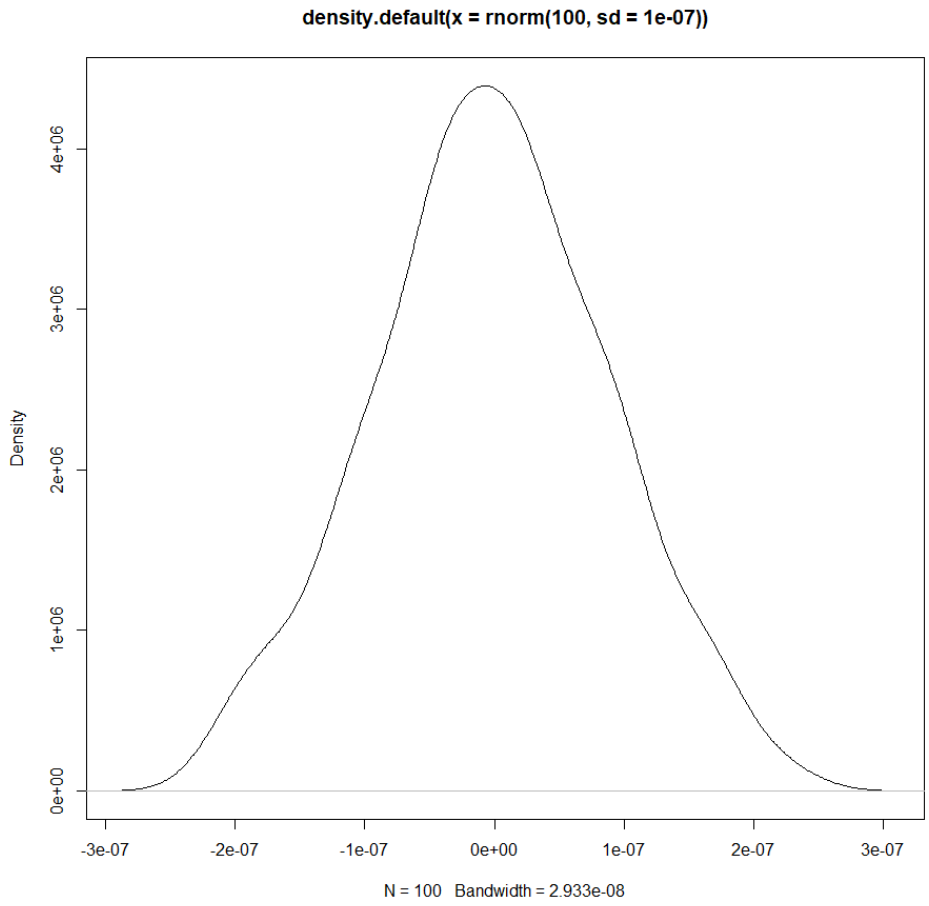
Desarrollo

Análisis de vectores virtuales

El primer fragmento de la transformación corresponde a un valor aleatorio de variación más el coeficiente de dilatación medio.

```
> plot(density( rnorm(100, sd = 0.00000005) + 0.00000115 ))
```

Se graficará directamente y entregará una representación similar a esta:



Esta gráfica de densidad de probabilidad nos muestra los diferentes valores de coeficientes de dilatación que podría tener cada medición; en otras palabras, es la parte que diversificará los datos.

Tendremos un vector de dimensión cien.

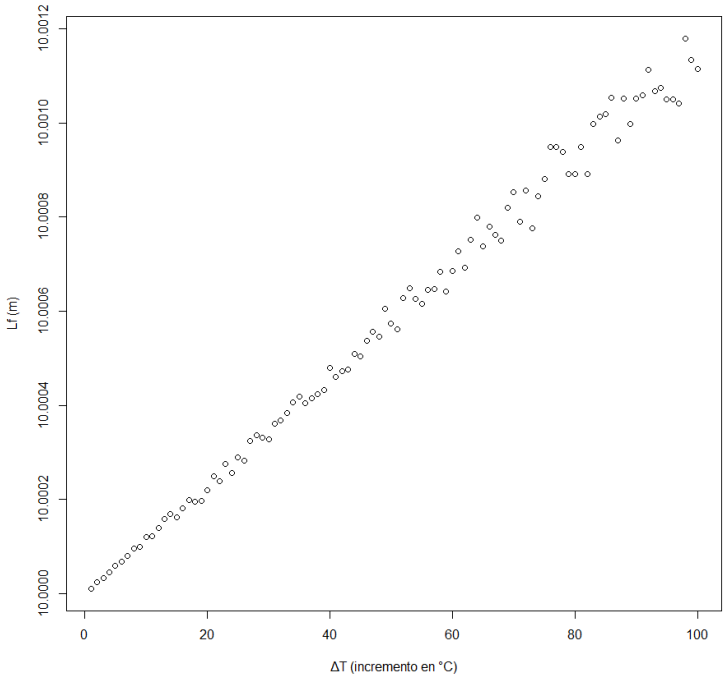
En la segunda parte de la transformación, se multiplicará por el delta de temperatura al que será sometida la varilla virtual.

Posteriormente, se aumentará a ese vector en una unidad. Esto servirá para multiplicar la longitud inicial por la unidad más el vector de proporciones de aumento de longitud.

El último paso es multiplicarlo por la longitud inicial de nuestra varilla, definida desde el inicio como diez metros.

Una vez creado el vector que contiene los datos transformados, se hará una gráfica para verificar la certidumbre de esta simulación.

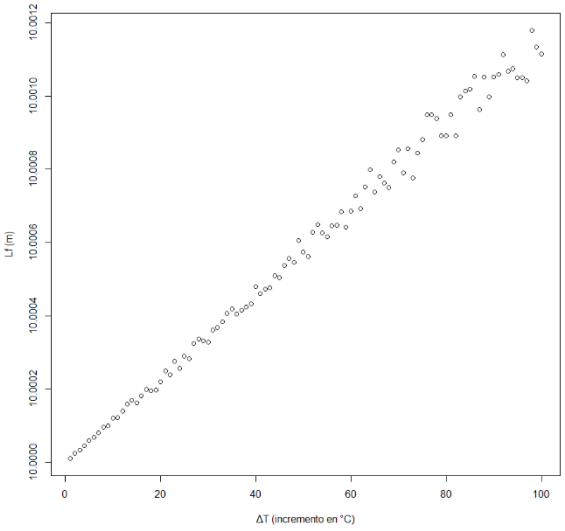
```
> plot(x,y,xlab = 'ΔT(incremento en °C)', ylab = 'Lf(m)')
```



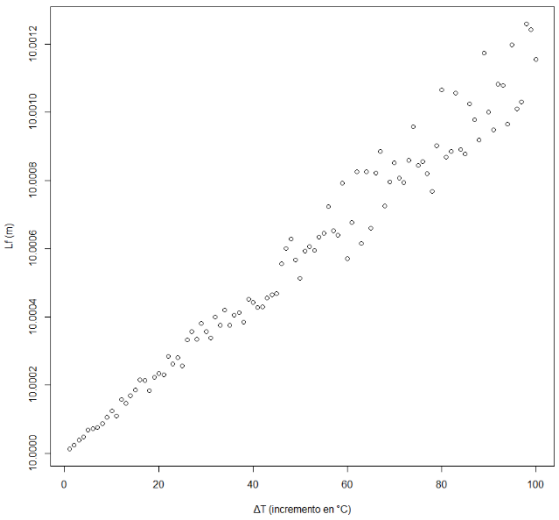
Inmediatamente, se observará una relación lineal entre las dos magnitudes, temperatura y longitud.

Posteriormente, se modificará la transformación para que alcance una desviación estándar mayor, por ejemplo 1 x 10^{-7}. Claramente se verá un aumento en la dispersión de los puntos en la gráfica.

```
> y <- 10 * (1 + (( rnorm(100, sd = 0.00000005) + 0.00000115 ) * x))
```



sd = 0.00000005



sd = 0.0000001

Conclusión

En el mundo del análisis de datos encontrar relaciones es muy importante para generar información y poder predecir ciertos fenómenos pues es éste el cimiento de la mayoría de las ciencias.

En esta práctica se usó R como herramienta para analizar información simulada de manera rápida. Lo que puede extrapolarse emplearse con información real.

Como Ingenieros de Datos, es importante contar con herramientas que permitan realizar tareas de la manera más rápida. R provee de estrategias que resultan más sencillas y con un mayor potencial que las técnicas usuales de los procesadores de hojas de cálculo, como Excel.

Ejercicio

Para crear una simulación de datos con una relación lineal basta conocer las coordenadas de un par de puntos del modelo matemático y la desviación estándar.

Genere un conjunto de datos que relacionen la edad en meses de un niño y su estatura con una población de 1000 individuos.

Considere sólo dos puntos de datos [(4, 63.9), (24, 87.8)] para edad (meses) y estatura (cm) respectivamente con una desviación estándar de 2.214.

Práctica 5. Regresión lineal

Introducción

Se ha generado previamente la virtualización de datos en la Práctica 4, tomando como ejemplo el fenómeno físico de la dilatación lineal experimentada por algunos materiales, evidenciado con el cambio de longitud la respuesta a las variaciones de temperatura.

Se empleará la función 'rnorm' y el modelo matemático de la ecuación lineal para la generación de datos semi aleatorizados a partir de un ejemplo hipotético de una varilla de 10 m, con coeficiente de dilatación de 1.1 $x10^{-5}$ $°C^{-1}$ y una desviación estándar de 5 x 10^{-8}. Además, se estudia la regresión lineal. Con el objetivo de explicar la relación que existe entre una variable dependiente y otra independiente.

Para identificar si un modelo de regresión lineal tiene sentido, se comienza con la realización de una gráfica de dispersión, misma que se generó en la Práctica 4.

Posteriormente, con la ayuda de R, se empleará el método de mínimos cuadrados que consiste en minimizar la suma de los cuadrados de los errores. Geométricamente, un error es la distancia entre el punto y la línea que traza el modelo lineal.

La importancia de conocer el coeficiente de regresión radica en que, con él, se genera una idea más certera de qué tan alejado es un modelo respecto a la realidad. Entre más pequeño sea el coeficiente más cerrado es el modelo de predicción, esto respecto a la perspectiva de los datos se tienen.

Objetivo

Hacer explícita la manera en que se realiza una regresión lineal en R, con la comparación del modelo de dilatación lineal generado en la práctica 4 con el modelo original del cual fueron obtenidos los mismos datos. Esto para comprobar la bilateralidad que R ofrece ante el tratamiento de los datos.

Desarrollo

Regresión Lineal

En R existe una función que computa directamente el modelo lineal, en referencia al promedio de errores más pequeños de cada modelo posible. A este error se le conoce como error estándar residual o error cuadrático medio, es una medida de la dispersión de los residuos en un modelo de regresión. Los residuos son las diferencias entre los valores observados y los valores predichos por el modelo. La ecuación para calcular el EER o ECM es la siguiente:

$$\text{ECM} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

ECM = sqrt(SCR / (n - p - 1))

Donde:

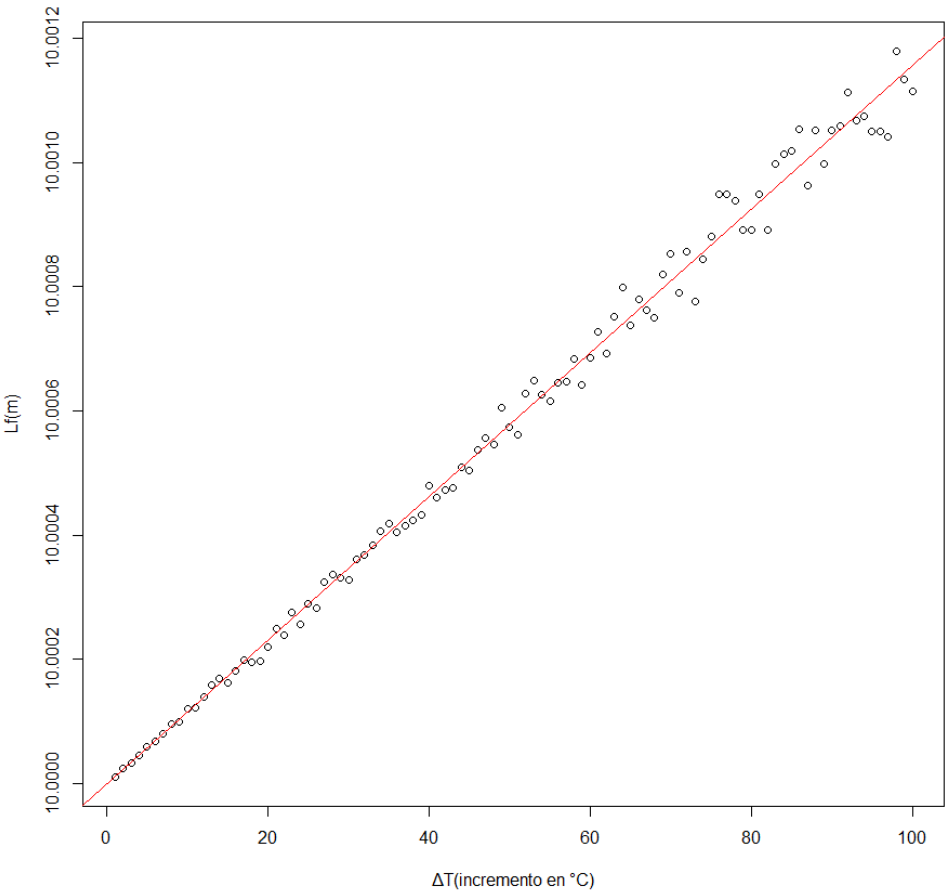
- SCR es la suma de los cuadrados de los residuos, es decir, la suma de las diferencias al cuadrado entre los valores observados y los valores predichos por el modelo.
- n es el número de observaciones en el conjunto de datos.
- p es el número de predictores o variables independientes en el modelo de regresión.

```
> plot(x,y,xlab = 'ΔT(incremento en °C)', ylab = 'Lf(m)')

> abline(lm(formula = y ~ x), col="red")
```

Como se mencionó con anterioridad, se retomarán los datos generados en la Práctica 4.

Se volverán a graficar los puntos, agregando una línea a nuestra gráfica que aproxima el modelo lineal más eficiente. Para esto, se usará la función lm(), la cual recibe un argumento tal que, de manera coloquial, esta parte ‘y ~ x’ se lee: “‘y’ dado ‘x’”. El resultado de esta función será transferido a abline(), para dibujar una línea de A a B dentro de un contexto (gráfica previa).



En el análisis de datos es tan importante la obtención de información de gráficas como de valores específicos. Para poder saber más sobre el modelo lineal se usará la función summary() que brinda los valores complementarios para generar datos que contribuyan a mejorar la comprensión del modelo.

```
> sm <- summary(lm(formula = y ~ x))

> sm
```

Dentro de la lista de vectores que sería 'sm' hay uno que resulta de especial interés al enfocarse en el rendimiento. Este es el vector de residuos, pues ayuda a evaluar el rendimiento del modelo lineal.

Conclusión

La verificación de los procesos realizados día a día contribuye a la autoevaluación y, por consiguiente, la identificación de nuestras fallas. Conocer los errores da la ventaja del diseño de estrategias para mejorar los procesos.

De igual manera, el poder realizar un proceso de manera directa e inversa mediante la ayuda de un lenguaje de programación como R genera retroalimentación inmediata para la verificación de los análisis.

Además, garantiza el dominio de ambos procesos de manera independiente y la experiencia para los datos empleando las herramientas estudiadas.

Ejercicio

Recordado el cálculo del error cuadrático medio, es hora de usar R para evaluarlo.

Encuentre el valor del error cuadrático del modelo; luego, altere la desviación estándar de la simulación de datos, vuelva a calcular y compare ambos valores. ¿Qué puede concluir de ello?

Glosario

Centroide

En matemáticas y en física el centroide es la intersección de todos los híper planos que dividen la figura en dos partes las cuales tienen n-volumen igual al del híper plano, esto puede simplificarse como: es la intersección de las medianas de un triángulo.

QGIS

Es un Sistema de Información Geográfica (SIG) de Código Abierto licenciado bajo GNU - General Public License, es un proyecto oficial de Open Source Geospatial Foundation (OSGeo) que corre sobre Linux, Unix, Mac OSX, Windows y Android y soporta numerosos formatos y funcionalidades de datos vector, datos ráster y bases de datos.

Ranura

Los objetos en R se componen de varios ‘slots’ o ranuras como los llamamos en esta práctica. Dentro del ambiente de desarrollo de R estos objetos se pueden componer de varias de estas ranuras que describen un subtipo de objeto dentro del objeto padre. Como ejemplo se usaron los objetos tipo datashape, los cuales tienen una ‘ranura’ que contiene el dataframe de las coordenadas de cada punto.

También se tiene el de ‘data’ que se usó en la manipulación de la información y para hacer uniones de manera natural, cada registro en las múltiples ranuras está hilado a las otras mediante índices, así vuelve el objeto relacional y manipulable.

Shapefile

El formato **ESRI Shapefile** (SHP) es un formato de archivo informático propietario de datos espaciales desarrollado por la compañía ESRI, quien crea y comercializa

software para Sistemas de Información Geográfica como Arc/Info o ArcGIS. Originalmente se creó para la utilización con su producto ArcView GIS, pero actualmente se ha convertido en formato estándar de facto para el intercambio de información geográfica entre Sistemas de Información Geográfica por la importancia que los productos ESRI tienen en el mercado SIG y por estar muy bien documentado.

Anexo 1. Librerías importadas

ggmap

ggmap es una paquetería de R la cual extiende las capacidades de gráficas de la paquetería desarrollada por Google llamada ggplot2.

Esta paquetería (ggmap) le brinda capacidades geoespaciales; es decir, contiene funciones que reciben un objeto geométrico y crea mapas con capacidades funcionales para su mejor despliegue.

rgdal

Provee una unión a la librería de abstracción geoespacial de Frank Warmerdam (GDAL) para su uso junto con R; por lo tanto, se tiene que preparar el ambiente de desarrollo para que soporte GDAL previamente a instalar esta paquetería.

Entre las funciones destacables dentro de esta paquetería se encuentra readOGR() que importa varios tipos de formatos de bases de datos geoespaciales y las convierte en un objeto geoespacial para que R lo entienda y manipule.

Por lo general, las funciones de rgdal aceptarán como datashape un archivo .shp el cual es un archivo con datos geométricos y viene al par con otros 3 archivos:

- .shx : contiene el mismo cabezal que el archivo shp pero su función es ser un índice del archivo shp.
- .dbf : en este archivo se almacenan los atributos de cada polígono del archivo geoespacial shp.
- .sbn : este es un índice geoespacial propietario por Esri.

rgeos

Es una interfaz al motor geométrico open source GEOS, usa una API de C para las operaciones topológicas a las geometrías que importaremos.

Esta serie de funciones son las que permiten hacer minería espacial pues contiene funciones compartidas en otros sistemas, como posGIS.

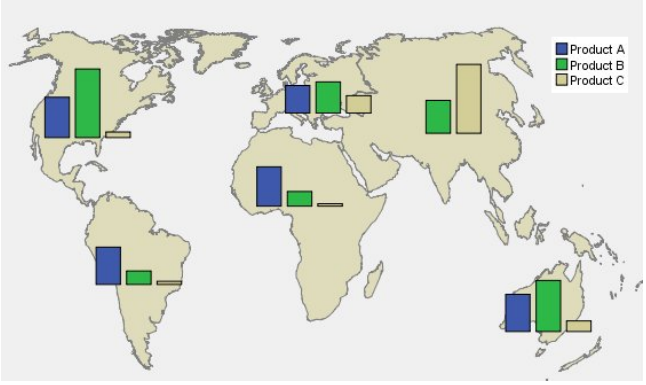
Regex

Es la manera de llamar a las expresiones regulares. En ciencia computacional se llama expresiones regulares a una cadena de caracteres que forman un patrón de búsqueda, estas expresiones regulares están basadas en la ciencia de lenguajes formales.

tmap

Los mapas temáticos son mapas geográficos en los que se visualizan las distribuciones de datos espaciales.

Esta paquetería ofrece un enfoque flexible, basado en capas y fácil de usar para crear mapas, tales como coropletas y mapas de burbujas.



Conclusión general

El resultado esperado es que el alumno conozca aquellas herramientas usadas actualmente para la exploración de datos y creación de información especializada.

Al final la ejecución de este compendio de prácticas, el estudiante debería:

- Conocer el paradigma de programación funcional.
- Reconocer la sintaxis de R.
- Generar gráficas dinámicas.
- Describir la Gramática de gráficas.
- Importar bases de datos a objetos de R.
- Reconocer y limpiar de inconsistencias los datos en R.
- Sub seleccionar dataframes.
- Importar archivos geoespaciales a R.
- Graficar mapas y personalizarlos.
- Usar funciones de agrupación.
- Generar información a partir de las herramientas de visualización.

Estos conocimientos serán base para que el alumno, junto con su curiosidad y necesidades laborales, pueda entrar a la profesión o investigación de la Ciencia de Datos.

Referencias

(2023) RStudio Desktop. https://www.rstudio.com/products/rstudio/download/

Becerril, Antonio. (2019, 14 Abril). La ciencia de datos gana terreno en la educación superior mexicana. El Economista. *https://www.eleconomista.com.mx/tecnologia/La-ciencia-de-datos-gana-terreno-en-la-educacion-superior-mexicana-20190414-0008.html*

Diagrama de Caja y Bigotes. Estadística para todos. *http://www.estadisticaparatodos.es/taller/graficas/cajas.html*

Robinson, Michael. (2020, 18 Septiembre). Administrative Level 1 Boundaries of Mexico. Mexico State Outlines. *https://www.arcgis.com/home/item.html?id=b8600a5a4054453f8a20e4999e9851d8*

Instituto Autónomo Tecnológico de México. ITAM. (2023,21 Abril) *The Comprehensive R Archive Network*. *https://cran.itam.mx*

Instituto Nacional de Estadística y Geografía. INEGI. Bienestar subjetivo - BIARE Básico

https://www.inegi.org.mx/investigacion/bienestar/basico/default.html

NumPy. (2023, 10 Mayo). The fundamental package for scientific computing with Python. https://numpy.org/#

Pandas. (2023). About Pandas. https://pandas.pydata.org/about/index.html

O'Neil, Cathy, Schutt, Rachel (2013) *Doing Data Science. O'Reilly Media, Inc.*

The R Foundation. (2023). What is *R*? https://www.r-project.org/about.html

Martínez, José. (2020, 10 Octubre) IArtificial.net Error Cuadrático Medio para Regresión

https://www.iartificial.net/error-cuadratico-medio-para-regresion/