Reporte Trabajo de Investigación

Análisis Masivo de datos - Programación Estadística I

Nicolás Molina González

2023-06-13

Introducción

El presente trabajo completa las actividades en la materia de Análisis Masivo de Datos de la Carrera de Estadística con la aplicación de conceptos aprendidos a lo largo del semestre relacionados con el manejo de bases de datos de tamaño apreciable, así como el manejo y presentación de la información usando el lenguaje R y los recursos que provee la librería Shiny.

Objetivos

1. Elegir una base de datos con información que pueda ser procesada y con información relevante (Bases de datos con al menos 10 mil observaciones y 10 variables).
2. Aplicar los conocimientos adquiridos en la materia para generar una aplicación web con la cual explorar la información de manera visual.
3. Generar una aplicacion web usando el framework Shiny para facilitar la visualizacion de al menos 10 variables incluidas en la Encuesta de Hogares del INE, relacionadas con las caracteristicas de la vivienda y acceso a servicios a nivel Bolivia.

Motivación

Enfrentar el problema de procesar información y presentarla de forma atractiva y completa para que sea útil para un potencial usuario final.

Marco Teórico

Shiny: Es un marco de trabajo para desarrollar aplicaciones web usando código en R. Está diseñado principalmente para científicos de datos y permite la creación de aplicaciones complejas sin un conocimiento previo de CSS, HTML o JavaScript. El paquete se encuentra actualmente en su versión 1.7.4 y se puede instalar e invocar en R usando los comandos:

```
install.packages("shiny")

library(shiny)
```

Las aplicaciones Shiny están contenidas en un script llamado `app.R` ubicado en un directorio (por ejemplo, `nuevodir/`) y la aplicación se puede ejecutar con el comando `runApp("nuevodir")` .

`app.R` tiene tres componentes:

- un objeto de interfaz de usuario
- una función servidor
- una llamada a la función `shinyApp`

El objeto de interfaz de usuario (`ui`) controla la disposición y apariencia de la aplicación. La función `server` contiene las instrucciones que necesita la aplicación para procesar la información. Finalmente, la función `shinyApp` crea los objetos de la aplicación Shiny a partir de un par explícito UI/server.

Cada aplicación Shiny tiene la misma estructura: un archivo `app.R` que contiene secciones `ui` y `server` , de la siguiente forma:

```
library(shiny)

ui <- ...

server <- ...

shinyApp(ui = ui, server = server)
```

Se recomienda que cada aplicación tenga su propio directorio.

Se puede ejecutar una aplicación Shiny dando el nombre de su directorio a la función `runApp` , por ejemplo si la aplicación se encuentra en un directorio llamado `mi_app` , se la ejecuta con el siguiente código:

```
library(shiny)
runApp("mi_app")
```

Shiny usa la función `fluidPage` para crear una pantalla que se ajusta automáticamente a las dimensiones de la ventana del navegador del usuario. La interfaz del usuario de la aplicación se construye ubicando elementos dentro de esta función, de la siguiente forma:

```
ui <- fluidPage(
  titlePanel("Título del panel"),

  sidebarLayout(
    sidebarPanel("Panel lateral"),
    mainPanel("Panel principal")
  )
)
```

Se puede crear disposiciones más avanzadas incluyendo interfaces de múltiples páginas que pueden incluir pestañas. Además es posible distribuir los elementos de la interfaz de acuerdo a un sistema de grilla (por filas y columnas).

Shiny soporta también la adición de contenido avanzado, por medio de funciones para incluir controles de interfaz de usuario, etiquetado HTML, formateo de texto, imágenes, etc.

Para dar contenido a los objetos incluidos en la interfaz de usuario se utiliza la función `server`, con ella se construye objetos `output` que contienen todo el código necesario para actualizar los objetos en la aplicación.

Cada llamada a `output` debería contener la salida de una de las funciones `render` de Shiny, entre las principales se tiene:

- `renderTable` para procesar estructuras tipo tabla (matrices, dataframes)
- `renderText` para crear cadenas de caracteres
- `renderPlot` para crear gráficos

Cada función `render` toma un sólo argumento: una expresión en R delimitada por llaves, `{}`. La expresión puede ser algo simple o abarcar múltiples líneas de código. Shiny usa estas instrucciones en `server` cuando se ejecuta la aplicación por primera vez y cada vez que necesita actualizar algún objeto.

Descripción de la base de datos

Para realizar el trabajo se ha elegido la base de datos de la Encuesta de Hogares. De la página del catálogo ANDA (https://anda.ine.gob.bo/index.php/catalog/93 (https://anda.ine.gob.bo/index.php/catalog/93)):

La Encuesta de Hogares (EH) realizada por el Instituto Nacional de Estadística (INE) el segundo semestre de cada gestión, tiene el propósito de generar información estadística de las principales características demográficas y socioeconómicas de la población boliviana, y se constituye, en la principal fuente de información de la medición de la pobreza por ingresos en Bolivia. De este modo, los resultados de la EH 2021 coadyuvan a la generación de información estadística necesaria para la formulación, evaluación, seguimiento de políticas y diseño de programas de acción contenidas en el marco del Plan de Desarrollo Económico y Social (PDES) y los Objetivos de Desarrollo Sostenible (ODS).

Entre los objetivos específicos de la EH 2021 se encuentran:

a. medir oportunamente el comportamiento de los indicadores de pobreza monetaria de la población boliviana en función a sus factores determinantes,
b. proporcionar información sobre las condiciones demográficas y socio económicas de la población y
c. producir estadísticas e indicadores sectoriales para el seguimiento de los resultados esperados del PDES y las metas de los ODS.

La definición del contenido temático de la EH 2021 recupera el planeamiento de las EH anteriores a la pandemia. En 2020, la EH planteó una boleta abreviada, debido a la reducción del cuestionario como estrategia para garantizar la recolección de información en el contexto de la pandemia por el COVID-19.

El alcance temático de la EH 2021 esta dado por:

a. Características generales del hogar y sus miembros: Caracteristicas socio demográficas y migración
b. Características en Salud: Salud general, fecundidad, enfermedades diarreicas agudas e infecciones respiratorias agudas, estilo de vida, seguridad ciudadana
c. Características Educativas: Formación educativa, causas de inasistencia, uso individual de TIC
d. Empleo: Condicion de actividad, ocupacion y actividad principal, ingresos del trabajador asalariado e independiente, actividad secundaria, ingreso laboral de la ocupación secundaria
e. Ingresos no laborales del hogar: Ingresos por transferecias, remesas
f. Defunciones en el hogar
g. Características de la vivienda particular
h. Acceso a la alimentacion en los hogares
i. Gastos: En alimentación dentro del hogar, gastos del hogar, equipamiento del hogar
j. Discriminación y seguridad ciudadana

UNIDAD GEOGRÁFICA

La cobertura geográfica de la Encuesta de Hogares 2021 es a nivel nacional. La información es recolectada en los nueve departamentos del país, tanto en área urbana como rural, a partir de un diseño de muestra previamente determinado.

NIVEL DE DESAGREGACIÓN

La desagregación es a nivel nacional, nacional urbano, nacional rural, y a nivel departamental, excepto Beni y Pando cuyas estimaciones son obtenidos de forma conjunta.

UNIDAD DE ANÁLISIS

El hogar y sus miembros del hogar.

UNIVERSO

Hogares y personas (nacionales o extranjeros) que residen en viviendas particulares ocupadas en el territorio nacional.

Para los fines del trabajo, se usará los resultados obtenidos para el componente g) de la encuesta.

Metodología

1. Tratamiento sobre la base de datos

La base de datos se encuentra en un archivo en formato SPSS (EH2021_Vivienda.sav). Para importarlo al ambiente de trabajo en RStudio se usan las librerías `haven` y `labelled`, con lo que se obtiene un dataframe (EH2021_Vivienda) de 12487 registros (filas) por 67 variables (columnas).

Dado que la encuesta contiene variables que no son relevantes para el analisis, se genera primero unas bases de datos auxiliares incluyendo solo la informacion que se procesara:

```
# Cargar los datos en la aplicacion
load("data/eh2021v.Rdata")

# Base de datos con las variables de analisis
bd <-  EH2021_Vivienda %>% select(depto, area, s07a_01, s07a_06, s07a_07, s07a_08, s0
7a_09, s07a_10, s07a_13, s07a_16, s07a_18, s07a_21, s07a_26, s07a_28)

# Base de datos usada para generar una tabla de variables
bdf <- bd %>% select(-depto, -area)
```

2. Indicadores y fichas de Indicadores

Para fines de visualizacion de la informacion sobre este censo se han elegido las siguientes variables:

1. s07a_01
2. s07a_06
3. s07a_07
4. s07a_08
5. s07a_09
6. s07a_10
7. s07a_13
8. s07a_16
9. s07a_18
10. s07a_21
11. s07a_26
12. s07a_28

Los detalles de cada una se muestran como resultado de la visualización.

3. Visualizacion de la información

Se ha elegido los graficos de la libreria `plotly` (https://plotly.com/r/ (https://plotly.com/r/)) y dos maneras de presentar la información para cada variable:

- Un grafico de torta para mostrar la distribucion porcentual de categorias, generado con el codigo:

```
plot_ly(bdf,
            labels = ~names(val_labels(bdf[[input$variable]])),
            values = ~summary(as.factor(bdf[[input$variable]])),
            type = 'pie') %>%
      layout(legend = l,
             paper_bgcolor='rgba(0,0,0,0)',
             plot_bgcolor='rgba(0,0,0,0)',
             title = "",
             xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALS
E),
             yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALS
E))
```

- Un grafico de treemap para mostrar la proporción absoluta de categorias, generado con el código:

```
plot_ly(
      bd,
      type= 'treemap',
      labels = names(val_labels(bd[[input$variable]])),
      parents = "",
      values = summary(as.factor(bd[[input$variable]])),
      textinfo="label+value+percent",
      domain=list(column=0)) %>%
        layout(paper_bgcolor='rgba(0,0,0,0)',
               plot_bgcolor='rgba(0,0,0,0)',
               title = "",
               margin = m)
```

4. Armado de la plataforma

Para organizar la interfaz de usuario de la aplicación se ha utilizado un tema visual provisto por la libreria `bslib` . El tema se carga despues de invocar la función `fluidPage` . Se agregó un titulo general para toda la pantalla con `titlePanel` y se usó la distribución de panel lateral y panel principal.

En el panel lateral `sidebarPanel` se coloca un control de opciones desplegable `selectInput` desde donde se puede elegir cualquiera de las variables bajo analisis. Debajo del control se incluye una tabla estatica `tableOutput` donde se ven los detalles de las variables que se puede analizar.

En el panel principal se incluyo un control de pestanas `tabsetPanel` con dos pestanas asignadas a la presentacion de los dos graficos usando el control `plotlyOutput` .

El código correspondiente se muestra a continuación:

```
ui <- fluidPage(

  theme = bslib::bs_theme(bootswatch = "superhero", navbar_bg = "#25443B"),
  titlePanel("Caracteristicas de viviendas en Bolivia, EH 2021"),

    sidebarLayout(

    sidebarPanel(
      selectInput("variable", "Variable", choices = names(bdf)),
      tableOutput('tblvar')
    ),

  mainPanel(
    textOutput("titulo_graf") %>% tagAppendAttributes(style= 'color: cyan;
                                                              font-size: 24px;
                                                              font-style: bol
d;'),
    tabsetPanel(
      id = "tabset",
      tabPanel("Porcentaje",

               plotlyOutput("torta_nac")),
      tabPanel("Cantidad",

               plotlyOutput("treemp"))
    )
  )
)
)
```

Resultados y análisis

- La aplicacion se ejecuta desde el ambiente de RStudio y puede correr tambien desde un navegador. Para hacer accesible la aplicacion a nivel global hay opciones como shinyapps (https://www.shinyapps.io/ (https://www.shinyapps.io/)) o correr un servidor de aplicaciones shiny desde una plataforma linux.
- Los controles de la interfaz responden a la interaccion del usuario y los graficos se actualizan con la informacion correspondiente.

Conclusiones y recomendaciones

- Shiny es una poderosa herramienta para desarrollo de aplicaciones web en ciencia de datos. Su facilidad inicial es aparente, ya que su extensibilidad permite abordar proyectos muy complejos, pero que a su vez se pueden volver dificiles de manejar.

- Para usar Shiny efectivamente se debe tener un alto grado de familiaridad con R y su ambiente de trabajo.
- La logica dinamica de Shiny para el manejo de informacion que provee el servidor usando simultaneamente datos capturados por la interfaz de usuario y luego actualizando elementos de la misma interfaz es un poco confusa al principio. El usuario inexperto debe modificar el enfoque lineal de ejecucion de procesos en R a un enfoque basado en la reaccion a interacciones.
- Un uso efectivo de Shiny como herramienta para los estadisticos y cientificos de datos en la carrera de estadistica de la UMSA requiere mucha practica adicional e idealmente guiada en el uso.