

Taller Análisis Exploratorio Estadístico del ICFES

Minería de Datos

Jonatan Camilo Igua Contreras

ID: 808919 | NRC: 73466

2025-04-0

Análisis Exploratorio resultados de la prueba saber 11

En el siguiente taller se realizo un análisis estadístico completo de los resultados del ICFES 2012.

1. Carga y exploración de los datos

Codificación dataset de ICFES: Para poder importar los datos del dataset del ICFES primero se tiene que mirar que tipo de codificación tiene el archivo .csv para saber como importarlo. Para esto se usa la funcion `guess_encoding()` del paquete `readr`, permite detectar la codificación de caracteres de un archivo CSV.

```
library(knitr)
library(dplyr)
library(readr)
guess_encoding("SB11-20121-RGSTRO-CLFCCN-V1-0-txt-csv.csv") %>%
  kable(col.names = c("Codificación", "Confianza"), caption = "Codificaciones detectadas")
```

Table 1: Codificaciones detectadas por `guess_encoding`

Codificación	Confianza
UTF-8	1.00
windows-1252	0.48

Codificación	Confianza
windows-1250	0.27

Como se puede apreciar se obtuvo las codificaciones sugeridas, la confidence indica la confianza del tipo de codificación en este caso el valor es 1.00 el cual es la máxima confianza, por ende el archivo se debe leer con la codificación UTF-8

Importar dataset de ICFES: Para importar los datos del archivo csv al proyecto de R se uso `read_csv` del paquete `readr`, para leer los datos y convertirlos en un data frame, se declaro que el tipo de codificación de los caracteres es UTF-8.

```
library(readr)
datos_icfes <- read_csv("SB11-20121-RGSTRO-CLFCCN-V1-0-txt-csv.csv", locale = loca
```

Despues de visualizar los pimeros datos importados en R se obtuvo un error Delimiter: “,” indicando que si bien los datos se importaron bien algunos caracteres se importaron un un formato erroneo por ese motivo se creo una segunda variable con el delimitador sep y se elimino el error. Para los analisis se tomo `datos_icfes_delimiter` para evitar errores con los datos.

```
datos_icfes_delimiter <- read_csv("SB11-20121-RGSTRO-CLFCCN-V1-0-txt-csv.csv", sep
```

¿Cuántas observaciones y variables hay? Para poder ver el numero de observaciones las filas de los datos se usa la función `nrow()`.

Table 2: Cantidad total de filas en el dataset

Número de filas
45390

Para poder ver el numero de variables las columnas se usa la función `ncol()`.

Table 3: Cantidad total de columnas en el dataset

Número de columnas
112

¿Que tipo de variables predominan: Numéricas o categóricas? Para poder determinar que tipos de variables predominan en el conjunto de datos de las notas

del ICFES, se uso la función `str()` para poder ver la estructura del data frame, esto permite ver el tipo de cada variable y su contenido. Se uso Kable para poder organizar los datos en formato de tabla.

```
#Primeros 20 elementos de los datos
estructura <- capture.output(str(datos_icfes_delimiter[ , 1:20]))
#Se guarda en la variable
estructura_df <- data.frame(estructura)

# Mostrar los datos
estructura_df %>%
  kable(col.names = c("Estructura de Datos"), caption = "Estructura del Dataset (p
```

Table 4: Estructura del Dataset (primeras 20 columnas)

Estructura de Datos
<p>‘data.frame’: 45390 obs. of 20 variables:</p> <p>\$ PERIODO : int 20121 20121 20121 20121 20121 20121 20121 20121 20121 20121 20121 ...</p> <p>\$ ESTU_TIPO_DOCUMENTO : chr “C” “C” “R” “C” ...</p> <p>\$ ESTU_PAIS_RESIDE : chr “CO” “CO” “CO” “CO” ...</p> <p>\$ ESTU_GENERO : chr “M” “M” “F” “M” ...</p> <p>\$ ESTU_NACIMIENTO_DIA : int 25 5 3 5 9 21 30 22 29 21 ...</p> <p>\$ ESTU_NACIMIENTO_MES : int 8 7 1 9 5 10 10 4 10 12 ...</p> <p>\$ ESTU_NACIMIENTO_ANNO : int 1992 1970 1994 1974 1972 1990 1984 1986 1994 1987 ...</p> <p>\$ ESTU_LIMITA_BAJAVISION : logi NA NA NA NA NA NA NA ...</p> <p>\$ ESTU_LIMITA_SORDOCEGUERA : chr “ ” “ ” “ ” “ ” ...</p> <p>\$ ESTU_LIMITA_COGNITIVA : chr “ ” “ ” “ ” “ ” ...</p> <p>\$ ESTU_LIMITA_INVIDENTE : chr “ ” “ ” “ ” “ ” ...</p> <p>\$ ESTU_LIMITA_MOTRIZ : chr “ ” “ ” “ ” “ ” ...</p> <p>\$ ESTU_LIMITA_SORDOINTERPRETE : chr “ ” “ ” “ ” “ ” ...</p> <p>\$ ESTU_LIMITA_SORDONINTERPRETE: chr “ ” “ ” “ ” “ ” ...</p> <p>\$ ESTU_ETNIA : int NA NA 1 NA NA NA NA NA NA NA ...</p> <p>\$ ESTU_COD_RESIDE_MCPPIO : int 5837 5837 5837 5837 5837 23001 5837 5837 5837 5837 ...</p> <p>\$ ESTU_RESIDE_MCPPIO : chr “TURBO” “TURBO” “TURBO” “TURBO” ...</p>

Estructura de Datos

```
$ ESTU_RESIDE_DEPTO : chr "ANTIOQUIA" "ANTIOQUIA"  
"ANTIOQUIA" "ANTIOQUIA" ...  
$ ESTU_ZONA_RESIDE : int 10 10 10 10 10 10 10 10 10 10 ...  
$ ESTU_AREA_RESIDE : int 1 1 1 1 1 1 1 2 1 1 ...
```

Utilizando la funcion `str()` sin limitar la cantidad de datos a mostrar se obtuvieron 45390 filas y 112 columnas .

Ver nombres y tipo de datos de las variables Con la funcion `sapply` podemos visualizar el nombre de las variables y su tipo de dato, se crea una tabla con dos columnas una con los nombres de las variables y otra con su tipo de dato.

```
library(knitr)  
  
tabla_tipos <- tibble::tibble(  
  Variable = names(datos_icfes_delimiter),  
  Tipo = sapply(datos_icfes_delimiter, class)  
)  
  
kable(tabla_tipos, caption = "Tipos de Variables en el DataFrame")
```

Table 5: Tipos de Variables en el DataFrame

Variable	Tipo
PERIODO	integer
ESTU_TIPO_DOCUMENTO	character
ESTU_PAIS_RESIDE	character
ESTU_GENERO	character
ESTU_NACIMIENTO_DIA	integer
ESTU_NACIMIENTO_MES	integer
ESTU_NACIMIENTO_ANNO	integer
ESTU_LIMITA_BAJAVISION	logical
ESTU_LIMITA_SORDOCEGUERA	character
ESTU_LIMITA_COGNITIVA	character
ESTU_LIMITA_INVIDENTE	character
ESTU_LIMITA_MOTRIZ	character
ESTU_LIMITA_SORDOINTERPRETE	character

Variable	Tipo
ESTU_LIMITA_SORDONINTERPRETE	character
ESTU_ETNIA	integer
ESTU_COD_RESIDE_MCPIO	integer
ESTU_RESIDE_MCPIO	character
ESTU_RESIDE_DEPTO	character
ESTU_ZONA_RESIDE	integer
ESTU_AREA_RESIDE	integer
IND_COD_ICFES_TERMINO	integer
COLE_COD_ICFES	integer
COLE_COD_DANE_INSTITUCION	numeric
COLE_NOMBRE_SEDE	character
COLE_CALENDARIO	character
COLE_GENERO	character
COLE_NATURALEZA	character
COLE_BILINGUE	character
COLE_JORNADA	character
COLE_CHARACTER	character
COLE_VALOR_PENSION	character
ESTU_VECES_ESTADO	character
ESTU_CARRDESEADA_TIPO	integer
ESTU_IES_COD_DESEADA	integer
ESTU_IES_COD_MPIO_DESEADA	integer
ESTU_IES_DEPT_DESEADA	character
ESTU_IES_DESEADA_NOMBRE	character
ESTU_IES_MPIO_DESEADA	character
ESTU_TOTAL_ALUMNOS_CURSO	character
ESTU_ANO_MATRICULA_PRIMERO	integer
ESTU_ANO_TERMINO_QUINTO	integer
ESTU_ANOS_COLEGIO_ACTUAL	integer
ESTU_ANO_MATRICULA_SEXTO	integer
ESTU_ANOS_PREESCOLAR	integer
ESTU_CUANTOS_COLE_ESTUDIO	integer
ESTU_REPROBO_CUARTO	integer
ESTU_REPROBO_DECIMO	integer
ESTU_REPROBO_NOVENO	integer
ESTU_REPROBO_OCTAVO	integer
ESTU_REPROBO_PRIMERO	integer
ESTU_REPROBO_QUINTO	integer
ESTU_REPROBO_SEGUNDO	integer

Variable	Tipo
ESTU_REPROBO_SEPTIMO	integer
ESTU_REPROBO_SEXTO	integer
ESTU_REPROBO_TERCERO	integer
ESTU_REPROBO_ONCE_MAS	integer
ESTU_POR_MEJORARPOSICIONSOCIAL	character
ESTU_POR_COLOMBIAAPRENDE	character
ESTU_POR_INFLUENCIAALGUIEN	character
ESTU_POR_INTERESPERSONAL	character
ESTU_POR_BUSCANDOCARRERA	character
ESTU_POR_TRADICIONFAMILIAR	character
ESTU_POR_ORIENTACIONVOCACIONAL	character
ESTU_RAZON_RETIRO	integer
ESTU_POR_AMIGOSESTUDIANDO	integer
ESTU_POR_COSTOMATRICULA	character
ESTU_POR_OPORTUNIDADES	character
ESTU_POR_OTRARAZON	character
ESTU_PRESTIGIOINSTITUCION	character
ESTU_POR_UBICACION	integer
ESTU_POR_UNICAQUEOFRECE	integer
ESTU_RETIRARSE_COLEGIO	character
ESTU_COD_MCPIO_PRESENTACION	character
ESTU_MCPIO_PRESENTACION	character
ESTU_DEPTO_PRESENTACION	character
ESTU_EXAM_NOMBREEEXAMEN	character
FAMI_EDUCA_PADRE	character
FAMI_EDUCA_MADRE	character
FAMI_OCUPA_PADRE	integer
FAMI_OCUPA_MADRE	integer
FAMI ESTRATO_VIVIENDA	integer
FAMI_NIVEL_SISBEN	integer
FAMI_PERSONAS_HOGAR	integer
FAMI_CUARTOS_HOGAR	integer
FAMI_PISOSHOGAR	integer
FAMI_TELEFONO_FIJO	integer
FAMI_CELULAR	integer
FAMI_INTERNET	integer
FAMI_SERVICIO_TELEVISION	integer
FAMI_COMPUTADOR	integer
FAMI_LAVADORA	integer

Variable	Tipo
FAMI_NEVERA	integer
FAMI_HORNO	integer
FAMI_DVD	integer
FAMI_MICROONDAS	integer
FAMI_AUTOMOVIL	integer
FAMI_INGRESO_FMILIAR_MENSUAL	integer
ESTU_TRABAJA	integer
ESTU_HORAS_TRABAJA	integer
PUNT LENGUAJE	integer
PUNT_MATEMATICAS	integer
PUNT_C_SOCIALES	integer
PUNT_FILOSOFIA	integer
PUNT_BIOLOGIA	integer
PUNT_QUIMICA	integer
PUNT_FISICA	integer
PUNT_INGLES	integer
DESEMP_INGLES	character
NOMBRE_COMP_FLEXIBLE	character
PUNT_COMP_FLEXIBLE	character
DESEMP_COMP_FLEXIBLE	character
ESTU_PUESTO	character

Análisis de las variables del dataset de ICFES Saber 11 Los tipos de datos encontrados en el dataset de las notas del ICFES son:

Character: Es el tipo de dato usado para texto o cadenas de caracteres.

Integer: Se usa para los números enteros sin decimales.

numeric: Se usa para los números decimales se puede usar para números enteros también.

logical: Se usa para valores booleanos TRUE o FALSE.

Los variables del conjunto de datos son sobre datos del estudiante, residencia del estudiante, información sobre el colegio, información sobre los grados cursados, información familiar e ingresos.

Cantidad de cada tipo de variable Para poder contar cada tipo de variable se uso la función `sapply()` la cual aplica una función a cada columna del data frame, se le aplico la función `class` para que tome el tipo de datos de cada variable, lo que

genera es un vector con los datos, con la función `table()` se cuenta cuantas veces aparece un valor del vector creado con `sapply()`.

```
tabla_tipos <- table(sapply(datos_icfes_delimiter, class))
tabla_tipos
```

```
##
## character    integer    logical    numeric
##          47         63         1         1
```

Los datos obtenidos son con el dataset original, sin haberle hecho ninguna técnica de transformación y limpieza de datos, por ende se van a tomar como datos cualitativos el tipo `character` y como datos cuantitativos los tipos: `integer` y `numeric`. Con este análisis en hay en total 47 variables categóricas o cualitativas y 64 variables cuantitativas o numéricas. Pero en el conjunto hay una variable con un tipo de dato `logical`.

```
variables_logicas <- names(datos_icfes_delimiter)[sapply(datos_icfes_delimiter, is
variables_logicas
```

```
## [1] "ESTU_LIMITA_BAJAVISION"
```

La variable contiene los siguientes datos:

```
unique(datos_icfes_delimiter$ESTU_LIMITA_BAJAVISION)
```

```
## [1] NA
```

```
all(is.na(datos_icfes_delimiter$ESTU_LIMITA_BAJAVISION))
```

```
## [1] TRUE
```

Con la función `unique` permite ver cuantos valores distintos hay el resultado fue `NA` osea esta variable no contiene datos y se uso `all` para verificar si en toda la variable soy hay datos `NA` y arrojo `TRUE` quiere decir que esta variable este campo nunca tuvo datos.

Conclusión ¿Que tipo de variables predominan? Después de conocer y analizar cada variable se puede concluir que hay mas variables de tipo numéricas con 63 variables y 47 de tipo categóricas por ahora se toman como categóricas, la variable `logical` se descarta ya que al no contar con información no es importante en este análisis de tipo de variables.

2. Limpieza y preparación de datos

Verificar valores faltantes En el anterior análisis se descubrió que la variable ESTU_LIMITA_BAJAVISION tiene valores faltantes, si se usa la función any con is.na se debe obtener un valor True.

```
any(is.na(datos_icfes_delimiter))
```

```
## [1] TRUE
```

Para contar cuantos Na hay en todo el conjunto de datos se usa la función sum().

```
sum(is.na(datos_icfes_delimiter))
```

```
## [1] 1318206
```

Se encontro un valor considerable de valores NA

Variables con valores NA Para ver los valores faltantes por columna se usa colSums() para poder ver de manera sencilla la cantidad de valores NA de cada variable.

```
#Por columna  
na_por_columna <- colSums(is.na(datos_icfes_delimiter))
```

```
# columnas con NA  
na_filtrados <- na_por_columna[na_por_columna > 0]
```

```
# Convertir a data frame  
naKable_df <- data.frame(  
  Variable = names(na_filtrados),  
  Cantidad_NA = as.numeric(na_filtrados)  
)
```

```
#tabla  
knitr::kable(naKable_df, caption = "Cantidad de valores faltantes (NA) por columna")
```

Table 6: Cantidad de valores faltantes (NA) por columna

Variable	Cantidad_NA
ESTU_NACIMIENTO_DIA	436
ESTU_NACIMIENTO_MES	436
ESTU_NACIMIENTO_ANNO	436
ESTU_LIMITA_BAJAVISION	45390
ESTU_ETNIA	42344
ESTU_COD_RESIDE_MCPIO	336
ESTU_ZONA_RESIDE	318
ESTU_AREA_RESIDE	39
IND_COD_ICFES_TERMINO	2660
COLE_COD_ICFES	4
COLE_COD_DANE_INSTITUCION	9526
ESTU_CARRDESEADA_TIPO	45268
ESTU_IES_COD_DESEADA	45281
ESTU_IES_COD_MPIO_DESEADA	45285
ESTU_ANO_MATRICULA_PRIMERO	45349
ESTU_ANO_TERMINO_QUINTO	45349
ESTU_ANOS_COLEGIO_ACTUAL	45349
ESTU_ANO_MATRICULA_SEXTO	45349
ESTU_ANOS_PREESCOLAR	45349
ESTU_CUANTOS_COLE_ESTUDIO	45355
ESTU_REPROBO_CUARTO	45349
ESTU_REPROBO_DECIMO	45349
ESTU_REPROBO_NOVENO	45349
ESTU_REPROBO_OCTAVO	45349
ESTU_REPROBO_PRIMERO	45349
ESTU_REPROBO_QUINTO	45349
ESTU_REPROBO_SEGUNDO	45349
ESTU_REPROBO_SEPTIMO	45349
ESTU_REPROBO_SEXTO	45349
ESTU_REPROBO_TERCERO	45349
ESTU_REPROBO_ONCE_MAS	45349
ESTU_RAZON_RETIRO	45380
ESTU_POR_AMIGOSESTUDIANDO	45350
ESTU_POR_UBICACION	45350
ESTU_POR_UNICAQUEOFRECE	45350
FAMI_OCUPA_PADRE	283
FAMI_OCUPA_MADRE	283

Variable	Cantidad_NA
FAMI_ESTRATO_VIVIENDA	2380
FAMI_NIVEL_SISBEN	289
FAMI_PERSONAS_HOGAR	297
FAMI_CUARTOS_HOGAR	41423
FAMI_PISOSHOGAR	354
FAMI_TELEFONO_FIJO	286
FAMI_CELULAR	45
FAMI_INTERNET	280
FAMI_SERVICIO_TELEVISION	282
FAMI_COMPUTADOR	282
FAMI_LAVADORA	282
FAMI_NEVERA	45
FAMI_HORNO	280
FAMI_DVD	282
FAMI_MICROONDAS	282
FAMI_AUTOMOVIL	282
FAMI_INGRESO_FMILIAR_MENSUAL	283
ESTU_TRABAJA	38100
ESTU_HORAS_TRABAJA	41676
PUNT LENGUAJE	78
PUNT_MATEMATICAS	4

Análisis de variables irrelevantes o vacías Después de mirar que variables contienen valores de tipo Na se analizo un conjunto de variables que para los siguientes análisis del taller no afectan a los resultados del análisis, estas variables son: Las variables de reprobación de grados: Se eliminan porque es una variable que no es relevante para los próximos análisis. Variables a eliminar: ESTU_REPROBO_CUARTO, ESTU_REPROBO_DECIMO, ESTU_REPROBO_NOVENO, ESTU_REPROBO_OCTAVO, ESTU_REPROBO_PRIMERO, ESTU_REPROBO_SEGUNDO. Se crea una nueva variable llamada datos_icfes_reprobo que guardara los datos sin tener en cuenta los que se van a descartar para los análisis.

```
library(dplyr)

datos_icfes_reprobo <- datos_icfes_delimiter %>% select(-c(
  ESTU_REPROBO_CUARTO,
  ESTU_REPROBO_DECIMO,
  ESTU_REPROBO_NOVENO,
```

```

    ESTU_REPROBO_OCTAVO,
    ESTU_REPROBO_PRIMERO,
    ESTU_REPROBO_QUINTO,
    ESTU_REPROBO_SEGUNDO,
    ESTU_REPROBO_SEPTIMO,
    ESTU_REPROBO_SEXTO,
    ESTU_REPROBO_TERCERO,
    ESTU_REPROBO_ONCE_MAS
  ))

```

Variables de Incapacidad: En el conjunto de datos se elimino la variable ESTU_LIMITA_BAJAVISION ya que no es relevante para el análisis del taller se eliminara de la variable datos_icfes_reprobo

```

datos_icfes_incapa <- datos_icfes_reprobo %>% select(-c(
  ESTU_LIMITA_BAJAVISION))

```

Valores inconsistentes o faltantes

ESTU_CUANTOS_COLE_ESTUDIO: Se analizo si hay datos inconsistentes en la cantidad de colegios de un estudiante se valida que si un estudiante tiene registrado mas de 10 colegios ya es un dato dudoso. Se encontraron ceros inconsistencias.

```

library(dplyr)

datos_icfes_incapa %>%
  filter(ESTU_CUANTOS_COLE_ESTUDIO > 10) %>%
  select(ESTU_CUANTOS_COLE_ESTUDIO) %>%
  distinct() %>%
  arrange(desc(ESTU_CUANTOS_COLE_ESTUDIO))

```

```

## [1] ESTU_CUANTOS_COLE_ESTUDIO
## <0 rows> (or 0-length row.names)

```

FAMI_PERSONAS_HOGAR = Se analiza esta variable en donde si hay 0 personas del hogar o mas de 20 seria un dato inconsistente. Se encontraron ceros inconsistencias.

```
datos_icfes_incapa %>%
  filter(FAMI_PERSONAS_HOGAR == 0 | FAMI_PERSONAS_HOGAR > 20) %>%
  select(FAMI_PERSONAS_HOGAR) %>%
  distinct() %>%
  arrange(FAMI_PERSONAS_HOGAR)
```

```
## [1] FAMI_PERSONAS_HOGAR
## <0 rows> (or 0-length row.names)
```

FAMI_ESTRATO_VIVIENDA: A la hora de analizar el estrado se encontraron inconsistencias con los datos ya que se encontraron estratos mayores del estrato 6

```
library(dplyr)
library(knitr)

# Tabla de frecuencias del estrato de vivienda (incluye NA)
datos_icfes_incapa %>%
  group_by(FAMI_ESTRATO_VIVIENDA) %>%
  summarise(Frecuencia = n()) %>%
  arrange(desc(Frecuencia)) %>%
  kable(caption = "Distribución de la variable FAMI_ESTRATO_VIVIENDA")
```

Table 7: Distribución de la variable FAMI_ESTRATO_VIVIENDA

FAMI_ESTRATO_VIVIENDA	Frecuencia
2	17500
1	12225
3	10337
NA	2380
4	2040
5	548
6	237
22	46
8	42
26	7
14	5
10	3

FAMI ESTRATO_VIVIENDA	Frecuencia
17	3
18	3
19	3
21	3
16	2
20	2
7	1
12	1
15	1
23	1

Como solución se decidió por filtrar y limpiar los valores mayores del estrato 6.

```
# Nuevo dataset, dejando solo estratos válidos
datos_icfes_estrato <- datos_icfes_incapa %>%
  mutate(FAMI ESTRATO_VIVIENDA = as.character(FAMI ESTRATO_VIVIENDA)) %>%
  filter(FAMI ESTRATO_VIVIENDA %in% c("1", "2", "3", "4", "5", "6") | is.na(FAMI_E
  mutate(FAMI ESTRATO_VIVIENDA = as.numeric(FAMI ESTRATO_VIVIENDA))

#Ver los estratos
library(dplyr)
library(knitr)

datos_icfes_estrato %>%
  group_by(FAMI ESTRATO_VIVIENDA) %>%
  summarise(Frecuencia = n()) %>%
  arrange(desc(Frecuencia)) %>%
  kable(caption = "Distribución nueva de la variable FAMI ESTRATO_VIVIENDA")
```

Table 8: Distribución nueva de la variable FAMI ESTRATO_VIVIENDA

FAMI ESTRATO_VIVIENDA	Frecuencia
2	17500
1	12225
3	10337
NA	2380

FAMI_ESTRATO_VIVIENDA	Frecuencia
4	2040
5	548
6	237

Estandarizar columnas con errores estructurales

Se estandarizo los datos de los documentos de los estudiantes, ya que los valores que contiene esta variable son:

```
## [1] "C" "R" "T" "E" "Q" "V" "P" ""
```

Para mejorar los datos se crea un vector con los nombres completos de los documentos y se usa `dplyr::recode` para remplazar los valores.

```
library(dplyr)

datos_icfes_limpios <- datos_icfes_estrato %>%
  mutate(ESTU_TIPO_DOCUMENTO = case_when(
    ESTU_TIPO_DOCUMENTO == "C" ~ "Cédula",
    ESTU_TIPO_DOCUMENTO == "P" ~ "Pasaporte",
    ESTU_TIPO_DOCUMENTO == "T" ~ "Tarjeta de Identidad",
    ESTU_TIPO_DOCUMENTO == "R" ~ "Registro Civil",
    ESTU_TIPO_DOCUMENTO == "E" ~ "Cédula de Extranjería",
    ESTU_TIPO_DOCUMENTO == "Q" ~ "Permiso Especial de Permanencia",
    ESTU_TIPO_DOCUMENTO == "V" ~ "Visa",
    ESTU_TIPO_DOCUMENTO == "" ~ "Sin Información",
    TRUE ~ ESTU_TIPO_DOCUMENTO # para conservar cualquier valor inesperado
  ))

unique(datos_icfes_limpios$ESTU_TIPO_DOCUMENTO)
```

```
## [1] "Cédula" "Registro Civil"
## [3] "Tarjeta de Identidad" "Cédula de Extranjería"
## [5] "Permiso Especial de Permanencia" "Visa"
## [7] "Pasaporte" "Sin Información"
```

Imputar datos faltantes Primera se usa la funcion `colSums` con `is.na` para ver las columnas que contiene valores NA del dataset limpiado.

```

library(knitr)
library(dplyr)

colSums(is.na(datos_icfes_limpios)) %>%
  as.data.frame() %>%
  tibble::rownames_to_column(var = "Variable") %>%
  rename(Valores_Faltantes = ".") %>%
  filter(Valores_Faltantes > 0) %>%
  kable(caption = "Cantidad de valores faltantes por variable")

```

Table 9: Cantidad de valores faltantes por variable

Variable	Valores_Faltantes
ESTU_NACIMIENTO_DIA	434
ESTU_NACIMIENTO_MES	434
ESTU_NACIMIENTO_ANNO	434
ESTU_ETNIA	42228
ESTU_COD_RESIDE_MCPIO	333
ESTU_ZONA_RESIDE	317
ESTU_AREA_RESIDE	38
IND_COD_ICFES_TERMINO	2622
COLE_COD_ICFES	3
COLE_COD_DANE_INSTITUCION	9508
ESTU_CARRDESEADA_TIPO	45226
ESTU_IES_COD_DESEADA	45163
ESTU_IES_COD_MPIO_DESEADA	45163
ESTU_ANO_MATRICULA_PRIMERO	45226
ESTU_ANO_TERMINO_QUINTO	45226
ESTU_ANOS_COLEGIO_ACTUAL	45226
ESTU_ANO_MATRICULA_SEXTO	45226
ESTU_ANOS_PREESCOLAR	45226
ESTU_CUANTOS_COLE_ESTUDIO	45232
ESTU_RAZON_RETIRO	45257
ESTU_POR_AMIGOSESTUDIANDO	45227
ESTU_POR_UBICACION	45227
ESTU_POR_UNICAQUEOFRECE	45227
FAMI_OCUPA_PADRE	282
FAMI_OCUPA_MADRE	282
FAMI_ESTRATO_VIVIENDA	2380

Variable	Valores_Faltantes
FAMI_NIVEL_SISBEN	283
FAMI_PERSONAS_HOGAR	296
FAMI_CUARTOS_HOGAR	41382
FAMI_PISOSHOGAR	282
FAMI_TELEFONO_FIJO	281
FAMI_CELULAR	44
FAMI_INTERNET	279
FAMI_SERVICIO_TELEVISION	281
FAMI_COMPUTADOR	281
FAMI_LAVADORA	281
FAMI_NEVERA	44
FAMI_HORNO	279
FAMI_DVD	281
FAMI_MICROONDAS	281
FAMI_AUTOMOVIL	281
FAMI_INGRESO_FAMILIAR_MENSUAL	282
ESTU_TRABAJA	38076
ESTU_HORAS_TRABAJA	41588
PUNT LENGUAJE	4

Con las variables con datos NA, se procedera a escoger algunas variables para realizar la imputacion de datos, la cual consiste en llenar los valores faltantes (NA) usando una tecnica logica en este caso usando la media, moda o diferentes tecnicas. Las variables que cuenten con una valor elevado de NA no seran imputadas por su dificultad, como lo son: ESTU_ETNIA, ESTU_CARRDESEADA_TIPO, ESTU_IES-COD-DESEADA etc Por otro lado hay variables como lo son: FAMI_OCUPA_PADRE, FAMI_OCUPA_MADRE, ESTU_TRABAJA, FAMI_CELULAR y FAMI_NEVERA que tienen pocos valores NA, se pueden imputar **Imputación Variables** Para las variables de tipo Integer como FAMI_OCUPA_PADRE se realiza la imputacion con la mediana la cual reemplaza los valores de NA por la media de los valores existentes, no se eliminan solo se reemplazan.

```
#Copia del dataframe
datos_icfes_limpios2 <- datos_icfes_limpios
#FAMI_OCUPA_PADRE con la mediana
datos_icfes_limpios2 $FAMI_OCUPA_PADRE[is.na(datos_icfes_limpios2$FAMI_OCUPA_PADRE)
  median(datos_icfes_limpios2$FAMI_OCUPA_PADRE, na.rm = TRUE)]
```

```
#La imputación se realizó correctamente
cat("Variable FAMI_OCUPA_PADRE imputada con mediana:\n")
```

```
## Variable FAMI_OCUPA_PADRE imputada con mediana:
```

```
summary(datos_icfes_limpios2$FAMI_OCUPA_PADRE)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   17.00   21.00   19.27   21.00   26.00
```

```
#FAMI_OCUPA_MADRE
```

```
datos_icfes_limpios2$FAMI_OCUPA_MADRE[is.na(datos_icfes_limpios2$FAMI_OCUPA_MADRE)] <-
  median(datos_icfes_limpios2$FAMI_OCUPA_MADRE, na.rm = TRUE)
```

```
cat("Variable FAMI_OCUPA_MADRE imputada con mediana:\n")
```

```
## Variable FAMI_OCUPA_MADRE imputada con mediana:
```

```
summary(datos_icfes_limpios2$FAMI_OCUPA_MADRE)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   19.00   22.00   19.98   22.00   26.00
```

```
#ESTU_TRABAJA integer
```

```
datos_icfes_limpios2$ESTU_TRABAJA[is.na(datos_icfes_limpios2$ESTU_TRABAJA)] <-
  median(datos_icfes_limpios2$ESTU_TRABAJA, na.rm = TRUE)
```

```
# La imputación se realizó correctamente
```

```
cat("Variable ESTU_TRABAJA imputada con mediana:\n")
```

```
## Variable ESTU_TRABAJA imputada con mediana:
```

```
summary(datos_icfes_limpios2$ESTU_TRABAJA)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000   1.000   1.000   1.133   1.000   7.000
```

```
#FAMI_CELULAR integer
```

```
datos_icfes_limpios2$FAMI_CELULAR[is.na(datos_icfes_limpios2$FAMI_CELULAR)] <-  
  median(datos_icfes_limpios2$FAMI_CELULAR, na.rm = TRUE)
```

```
# La imputación se realizó correctamente
```

```
cat("Variable FAMI_CELULAR imputada con mediana:\n")
```

```
## Variable FAMI_CELULAR imputada con mediana:
```

```
summary(datos_icfes_limpios2$FAMI_CELULAR)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## 0.0000  1.0000  1.0000  0.9437  1.0000  1.0000
```

```
#FAMI_NEVERA integer
```

```
datos_icfes_limpios2$FAMI_NEVERA[is.na(datos_icfes_limpios2$FAMI_NEVERA)] <-  
  median(datos_icfes_limpios2$FAMI_NEVERA, na.rm = TRUE)
```

```
# La imputación se realizó correctamente
```

```
cat("Variable FAMI_NEVERA imputada con mediana:\n")
```

```
## Variable FAMI_NEVERA imputada con mediana:
```

```
summary(datos_icfes_limpios2$FAMI_NEVERA)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
## 0.0000  1.0000  1.0000  0.8971  1.0000  1.0000
```

Las variables seleccionadas se imputaron correctamente por esta razón si se mira el data frame `datos_icfes_limpios2` estas no aparecen cuando se filtra por valores NA

```

library(knitr)
library(dplyr)

colSums(is.na(datos_icfes_limpios2)) %>%
  as.data.frame() %>%
  tibble::rownames_to_column(var = "Variable") %>%
  rename(Valores_Faltantes = ".") %>%
  filter(Valores_Faltantes > 0) %>%
  kable(caption = "Cantidad de valores faltantes por variable Imputado")

```

Table 10: Cantidad de valores faltantes por variable Imputado

Variable	Valores_Faltantes
ESTU_NACIMIENTO_DIA	434
ESTU_NACIMIENTO_MES	434
ESTU_NACIMIENTO_ANNO	434
ESTU_ETNIA	42228
ESTU_COD_RESIDE_MCPIO	333
ESTU_ZONA_RESIDE	317
ESTU_AREA_RESIDE	38
IND_COD_ICFES_TERMINO	2622
COLE_COD_ICFES	3
COLE_COD_DANE_INSTITUCION	9508
ESTU_CARRDESEADA_TIPO	45226
ESTU_IES_COD_DESEADA	45163
ESTU_IES_COD_MPIO_DESEADA	45163
ESTU_ANO_MATRICULA_PRIMERO	45226
ESTU_ANO_TERMINO_QUINTO	45226
ESTU_ANOS_COLEGIO_ACTUAL	45226
ESTU_ANO_MATRICULA_SEXTO	45226
ESTU_ANOS_PREESCOLAR	45226
ESTU_CUANTOS_COLE_ESTUDIO	45232
ESTU_RAZON_RETIRO	45257
ESTU_POR_AMIGOSESTUDIANDO	45227
ESTU_POR_UBICACION	45227
ESTU_POR_UNICAQUEOFRECE	45227
FAMI_ESTRATO_VIVIENDA	2380
FAMI_NIVEL_SISBEN	283

Variable	Valores_Faltantes
FAMI_PERSONAS_HOGAR	296
FAMI_CUARTOS_HOGAR	41382
FAMI_PISOSHOGAR	282
FAMI_TELEFONO_FIJO	281
FAMI_INTERNET	279
FAMI_SERVICIO_TELEVISION	281
FAMI_COMPUTADOR	281
FAMI_LAVADORA	281
FAMI_HORNO	279
FAMI_DVD	281
FAMI_MICROONDAS	281
FAMI_AUTOMOVIL	281
FAMI_INGRESO_FMILIAR_MENSUAL	282
ESTU_HORAS_TRABAJA	41588
PUNT LENGUAJE	4

2.1 Nuevas transformaciones

Calcular edad estudiante Para poder calcular edad del estudiante apartir de su fecha de nacimiento y la fecha del examen que es del 2012 se usan las variables: ESTU_NACIMIENTO_DIA, ESTU_NACIMIENTO_MES, ESTU_NACIMIENTO_ANNO.

Se creo una nueva columna con la fecha completa del estudiante llamada FECHA_NACIMIENTO

```
#Nueva columna con fecha completa
datos_icfes_limpios2$FECHA_NACIMIENTO <- as.Date(paste(datos_icfes_limpios2$ESTU_N
#Se toma una fecha del año 2012 ya que no esta la fecha exacta en los datos
fecha_examen <- as.Date("2012-01-01")

# Calcular la edad se restan la fecha de nacimiento de la fecha del examen
datos_icfes_limpios2$EDAD <- as.numeric(difftime(fecha_examen, datos_icfes_limpios
# Redondear las edades a enteros
datos_icfes_limpios2$EDAD <- round(datos_icfes_limpios2$EDAD)
library(knitr)
```

```
# Mostrar solo las primeras 15 filas con la columna "EDAD"
kable(head(datos_icfes_limpios2 %>% select(EDAD), 15),
      col.names = c("Edad"),
      caption = "Primeras 15 Edades")
```

Table 11: Primeras 15 Edades

Edad
19
41
18
37
40
21
27
26
17
24
23
20
17
18
21

Nueva columna edad categorizada Se creo una nueva columna llamada CATEGORIA_EDAD, para guardar las categorias de las edades, se uso la funcion mutate(), para poder agregar los valores en las columnas, y con el case_when() se crean las categorias de la edad.

```
# Crear la nueva columna "CATEGORIA_EDAD"
datos_icfes_limpios2 <- datos_icfes_limpios2 %>%
  mutate(CATEGORIA_EDAD = case_when(
    EDAD >= 12 & EDAD <= 17 ~ "Adolescente",
    EDAD >= 18 & EDAD <= 26 ~ "Joven",
    EDAD >= 27 & EDAD <= 59 ~ "Adulto",
    EDAD >= 60 ~ "Adulto mayor",
    TRUE ~ "Desconocido" # Para edades fuera de los rangos
  ))
```

```
library(knitr)

kable(head(datos_icfes_limpios2 %>% select(EDAD, CATEGORIA_EDAD), 15),
       col.names = c("Edad", "Categoría de Edad"),
       caption = "Primeras 15 Edades y Categorías de Edad")
```

Table 12: Primeras 15 Edades y Categorías de Edad

Edad	Categoría de Edad
19	Joven
41	Adulto
18	Joven
37	Adulto
40	Adulto
21	Joven
27	Adulto
26	Joven
17	Adolescente
24	Joven
23	Joven
20	Joven
17	Adolescente
18	Joven
21	Joven

3. Analisis Exploratorio y Univariado

** ## 4. Análisis Bivariado

¿Quien se destaca más en matemáticas hombres o mujeres?

¿Existe diferencia significativa en el puntaje global por género?

¿Qué tan fuerte es la relación entre matemáticas y lectura crítica?

5. Visualización socioeconómica