INFORME PROYECTO FINAL PROBABILIDAD Y ESTADÍSTICA

PunkerGhoul

RESUMEN: *En este informe, se dará la investigación estadística y probabilística de un Dataset, haciendo uso del lenguaje de programación R en la herramienta RStudio para analizar los datos recopilados implementando la estadística descriptiva. Al tiempo que es importante recalcar el interés desde la ingeniería en realizar el análisis de este tipo de datos de carácter social ya que mediante el análisis de estos datos se puede dar una mejor organización con mejor rendimiento en la generación de recursos para la educación de la humanidad y en la práctica de herramientas usadas con tal que se aumente la cantidad de ingenieros en sistemas u otras áreas que también se vean involucradas en los estudios vistos.*

PALABRAS CLAVE: Análisis, estadística, programación, tecnología.

1 INTRODUCCIÓN

Para comenzar hay que contextualizar sobre el Dataset, caya descripción es acerca la “Información de estudiantes y aspirantes. La ruta de aprendizaje 1 va dirigida a estudiantes de grados 9, 10 y 11 de colegios oficiales del país. Los estudiantes podrán escoger entre fundamentos de programación e industrias creativas digitales. La ruta de aprendizaje 2 va dirigida a los colombianos que se encuentren interesados en desarrollar en desarrollar habilidades en programación. Cuenta con un ciclo básico (programación básica, fundamentos de programación y desarrollo de software); y ciclo de profundización (desarrollo de aplicaciones web y desarrollo de aplicaciones móviles).” [2] con lo cual se quiere mantener un registro acerca las personas que se inscribieron en el programa MINTIC durante el año 2020 que están determinadas a empezar o seguir aprendiendo acerca la Tecnología, la Información y la Comunicación, esto reduciendo el alcance según a las ubicaciones de las personas jóvenes entre los 14 y 28 años que pertenezcan a un estrato medio o bajo. Esto con el propósito de representar el análisis estadístico y gráfico de los datos recopilados tal que en un futuro se pueda implementar un programa orientado a estas personas con un mejor apoyo de recursos tal que se disminuya la tasa de desertores.

2 PREPARACIÓN

Primero que todo, hay que descargar e importar los datos en la plataforma de desarrollo del lenguaje R.

$$\#\# \boldsymbol{Importar\ DataSet}$$
$$mydata <- read.csv2(file = "../Misi_n_TIC_2020_100_mil_programadores_CSV.csv", sep = ",", encoding = "UTF-8")$$

(1)

Se verifica que se hayan importado las columnas con el siguiente comando.

$$names(mydata)$$

(2)

El cual da como resultado los nombres de las columnas que contienen sus respectivos datos del Dataset importado.

```
 [1] "FECHA_CORTE"       "DEPARTAME_NOMBRE"  "MUNICIPIO_NOMBRE"  "ESTADO_FORMACION"  "ESTADOS"
 [6] "CONVOCATORIA"      "ESTRATO_SOCIAL"    "EDAD"              "GENERO"            "RUTA"
[11] "GRADO"             "META_BENEFICIARIO" "INVERSION"         "META_DINERO"       "FECHA_VIGENCIA"
```

Figura 1. Resultado columnas de base de datos.

Entonces al revisar los datos que contienen las columnas, se determina el proceso a seguir para filtrar los datos requeridos.

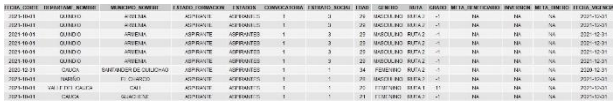

Figura 2. Resultado columnas de base de datos.

Esto se logra haciendo uso del siguiente comando con el que se filtran los datos según lo requerido.

$$mydata.filtered <- subset(mydata, (mydata\$EDAD > 13 \,\&\, mydata\$EDAD < 29) \,\&\, (mydata\$ESTRATO\_SOCIAL < 4))$$

(3)

De estos datos se es requerido ciertos conjuntos de datos, los cuales son la edad, el estrato, y la ubicación, la cual se divide en departamento y municipio. Los cuales pueden ser almacenados en constantes especificadas con el siguiente código.

$$mydata.filtered.edad <- mydata.filtered\$EDAD$$

$$mydata.filtered.estrato <- as.integer(mydata.filtered\$ESTRATO\_SOCIAL)$$

$$mydata.filtered.ubicacion <- paste(mydata\$MUNICIPIO_{NOMBRE}, mydata\$DEPARTAME_{NOMBRE}, sep = ',')$$

$$mydata.filtered.ubicacion.general <- mydata\$DEPARTAME\_NOMBRE$$

(4)

También hay que importar una librería con la cual se realizarán algunos análisis estadísticos, esto a través del siguiente código.

$$library(moments) \quad (5)$$

Al tiempo que también es necesario crear ciertas funciones que serán utilizadas más adelante con el mismo propósito, es decir para las medidas de tendencia central, las medidas de variación y la estadística descriptiva.

```
suma = function(x, y) {
  return(x + y)
}
```

Figura 3. Función de suma.

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

Figura 4. Función de moda.

```
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}
```

Figura 5. Función de normalización.

```
znormalizar <- function(x) {
  return((x - mean(x)) / sd(x))
}
```

Figura 6. Función que normaliza un vector según la distribución normal.

```
mgeometrica <- function(x) {
  n = length(x)
  producto = prod(x)
  return(producto ** (1 / n))
}
```

Figura 7. Función de media geométrica.

```
mponderada <- function(x, f) {
  if(length(x) == length(f)) {
    if(sum(f) == 1) return(x%*%f)
    else return("El vector de probabilidad debe sumar 1.0")
  }
  return("Los vectores deben tener el mismo tamaño")
}
```

Figura 8. Función de media ponderada.

```
marmonica <- function(x) { #También se puede usar 1/mean(1/x)
  n = length(x)
  sumarecip = sum(1 / x)
  return(n / sumarecip)
}
```

Figura 9. Función de media armónica.

```
rango <- function(v) {
  return(max(mydata.filtered.estrato) - min(mydata.filtered.estrato))
}
```

Figura 10. Función de rango.

```
estdescript = function(dataset) {
  print(paste("Media Aritmética:", mean(dataset)))
  print(paste("Mediana:", median(dataset)))
  print(paste("Moda:", getmode(dataset)))
  print(paste("Media Geométrica:", mgeometrica(dataset)))
  print(paste("Media Armónica:", marmonica(dataset)))
  print(paste("Rango:", rango(dataset)))
  print(paste("Rango Intercuartílico:", IQR(dataset)))
  print(paste("Varianza:", var(dataset)))
  print(paste("Desviación Estándar:", sd(dataset)))
  print(paste("Coeficiente de Asimetría:", skewness(dataset)))
  print(paste("Curtosis:", kurtosis(dataset)))
}
```

Figura 11. Función de estadística descriptiva.

3 DESARROLLO

Para dar inicio con el análisis de los datos, se usó el siguiente código con el cual respectivamente a la edad y los estratos de los aspirantes MINTIC 2022 se pudo obtener el mínimo, el primer cuartil, la mediana, el promedio, el tercer cuartil y el máximo, así como también se implementó el código que permita obtener un diagrama que resuma las características principales de los datos como posición, dispersión, asimetría, entre otros, tal que permita identificar la presencia de valores atípicos.

```
summary(mydata.filtered.edad, main = "Edad Aspirantes MINTIC 2022")
boxplot(mydata.filtered.edad, main = "Edad Aspirantes MINTIC 2022")

summary(mydata.filtered.estrato, main = "Estrato Aspirantes MINTIC 2022")
boxplot(mydata.filtered.estrato, main = "Estrato Aspirantes MINTIC 2022")
```

Figura 12. Bloque de código que resume datos.

Este código da como resultado en orden, los siguientes datos.

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  14.00   17.00   21.00   20.97   24.00   28.00
```

Figura 13. Resultado de la función summary aplicada en la edad.

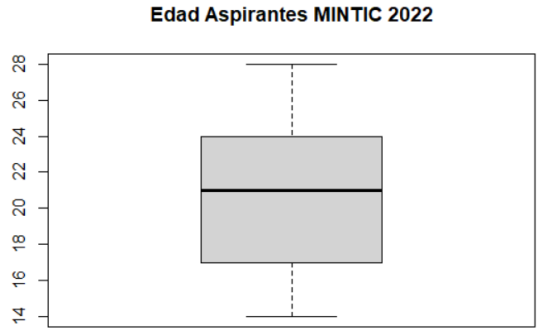

Figura 14. Diagrama de cajas de la edad.

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.000   1.000   2.000   1.967   3.000   3.000
```

Figura 15. Resultado de la función summary aplicada en el estrato.

7

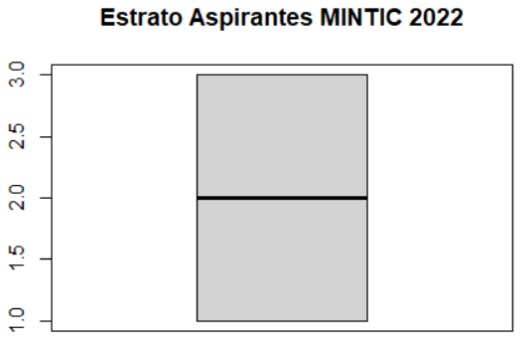

Figura 16. Diagrama de cajas del estrato.

Según los resultados se puede seguir con un análisis más específico en cada conjunto de datos.

4 CONJUNTOS DE DATOS

En esta sección se describirá cada conjunto de datos mediante un análisis estadístico en el que se encontrarán resultados en datos, gráficas, tablas y diagramas.

4.1 EDAD

Para la edad, según los datos filtrados, se podrá obtener una mejor visión de la situación presente según el objetivo.

4.1.1 CALCULO PORCENTUAL

Se usa el siguiente bloque de código para reunir los valores del conjunto de datos e inicializar el porcentaje que acumulará cada dato.

```
perc.edad = NULL
vals.edad = table(mydata.filtered.edad)
```

Figura 17. Acumulador de porcentaje y tabla del conjunto de datos.

Este bloque de código almacena cada cantidad de tipo de datos que contenga el conjunto de datos de edad, tal como se puede ver a continuación.

```
mydata.filtered.edad
   14    15    16    17    18    19    20    21    22    23    24    25    26    27
 1446  8029 18332 21134 17194 12711 11734 11173 11291 11564 11829 11899 11535 11355 10
```

Figura 18. Cantidad de cada tipo de dato que contiene el conjunto.

Posteriormente se almacenan los porcentajes correspondientes haciendo uso del siguiente código.

```
for (i in 1:length(vals.edad)) {
  perc.edad = round(cbind(perc.edad, 100 * vals.edad[[i]] / sum(vals.edad)), 1)
}
```

Figura 19. Bloque de código que añade los porcentajes en el acumulador.

El acumulador toma la siguiente forma.

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]  0.8  4.4 10.1 11.6  9.5    7  6.5  6.1  6.2   6.4   6.5   6.5   6.3   6.2   5.9
```

Figura 20. Porcentajes acumulados.

A continuación se establecen las etiquetas que aparecerán en los diagramas en los cuales se podrán apreciar los datos correspondientes.

$$labels.edad = paste(names(vals.edad), "AÑOS = ", perc.edad, "\%") \quad (6)$$

Es entonces que se usa el siguiente código con el propósito de representar los datos mediante un diagrama circular.

$$pie(vals.edad, labels = labels.edad, main = "Edad\ Aspirantes\ MINTIC\ 2022") \quad (7)$$

Este código permite graficar de la siguiente forma.

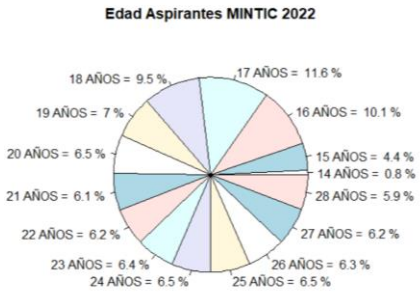

Figura 21. Diagrama circular de edades con su respectivo porcentaje.

De estos datos se puede evidenciar que aquellos aspirantes con más presencia entre los jóvenes pertenecen a aquellos con una edad de 17 años. Esto se puede observar con mayor facilidad desde un diagrama de barras.

$$barplot(vals.edad, main = "Edad\ Aspirantes\ MINTIC\ 2022") \quad (8)$$

Este código permite graficar de la siguiente forma.

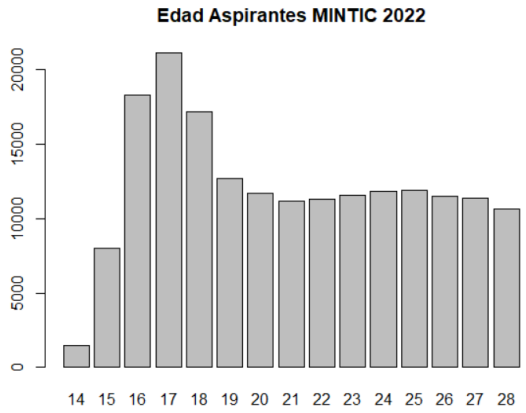

Figura 22. Diagrama de barras de edades.

De estos datos también es posible sacar el diagrama de tallo y hojas que representa la distribución entre los datos, haciendo uso del siguiente código.

$$stem(mydata.filtered.edad, scale = 1, width = 20, atom = 0.01) \quad (9)$$

Del cual resulta la siguiente tabla.

```
The decimal point is at the |

14 | 00000000+1426
15 | 00000000+8009
16 | 00000000+18312
17 | 00000000+21114
18 | 00000000+17174
19 | 00000000+12691
20 | 00000000+11714
21 | 00000000+11153
22 | 00000000+11271
23 | 00000000+11544
24 | 00000000+11809
25 | 00000000+11879
26 | 00000000+11515
27 | 00000000+11335
28 | 00000000+10656
```

Figura 23. Diagrama de tallo y hojas.

De este conjunto de datos se realiza el respectivo análisis estadístico haciendo uso del conjunto de las funciones dadas acumuladas en la función que será usada de la siguiente forma.

$$estdescript(mydata.filtered.edad) \quad (10)$$

El resultado de este código es mostrado de la siguiente forma.

```
[1] "Media Aritmética: 20.9714626557157"
[1] "Mediana: 21"
[1] "Moda: 17"
[1] "Media Geométrica: Inf"
[1] "Media Armónica: 20.2095623239637"
[1] "Rango: 2"
[1] "Rango Intercuartílico: 7"
[1] "Varianza: 16.2076561571724"
[1] "Desviación Estándar: 4.02587334092522"
[1] "Coeficiente de Asimetría: 0.214135240723099"
[1] "Curtosis: 1.76606670425145"
```

Figura 24. Resultados de análisis estadístico de edad.

Además es posible sacar la normalización de este conjunto de datos, la cual es dada entre el mínimo y el máximo, esto haciendo uso del siguiente código.

```
print("Normalización entre el máximo y mínimo:")
xnormal.edad = normalize(mydata.filtered.edad)
```

Figura 25. Bloque de código de normalización de datos de edad entre mínimo y máximo.

Este bloque de código da como resultado lo siguiente.

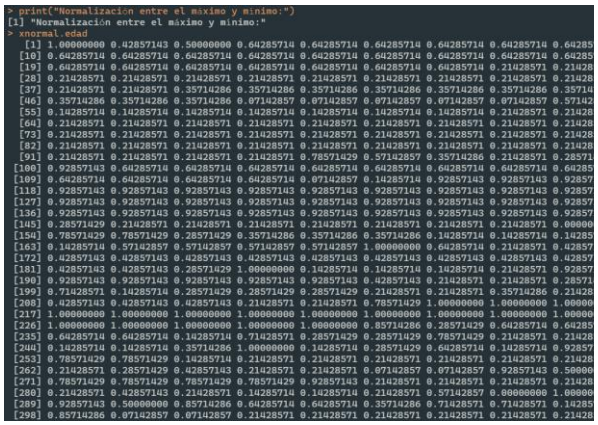

Figura 26. Parte de resultados de normalización de edad entre máximo y mínimo.

De la misma manera también es posible la normalización con distribución normal con el siguiente código.

```
print("Normalización con la Distribución Normal:")
xnormz.edad = znormalizar(mydata.filtered.edad)
```

Figura 27. Bloque de código de normalización de datos de edad con distribución normal.

Este bloque de código da como resultado lo siguiente.

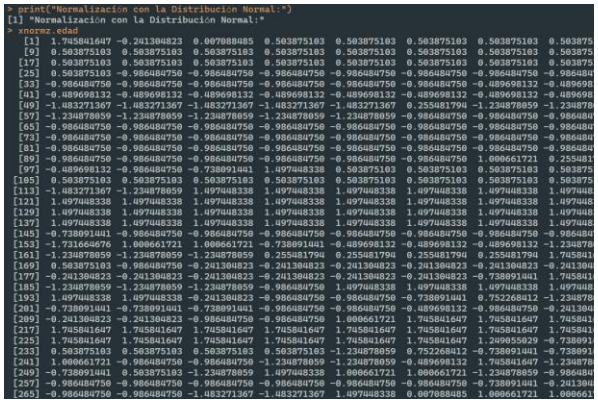

Figura 28. Parte de resultados de normalización de edad con distribución normal.

Para reunir los datos tal que se pueda dar como resultado la tabla de frecuencia del conjunto de datos de edad, se almacenan datos como los que se verán a continuación en los códigos. Empezando por las frecuencias relativas se utiliza el siguiente código.

$$frelativas.edad \ <- round(prop.table(vals.edad) * 100, 3) \quad (11)$$

Este da como resultado una tabla que almacena dichos datos, tal y como se ve a continuación.

Figura 29. Tabla de frecuencias relativas de edad.

Del mismo modo se obtiene la tabla de frecuencias absolutas acumuladas con el siguiente comando.

$$fabsol.acum.edad \ <- cumsum(vals.edad) \quad (12)$$

El cual da como resultado la siguiente tabla.

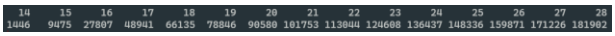

Figura 30. Tabla de frecuencias absolutas acumuladas de edad.

De estos datos es posible sacar la frecuencia relativa acumulada con el siguiente código.

$$frelat.acum.edad \ <- cumsum(frelativas.edad) \quad (13)$$

El cual da como resultado la siguiente tabla.

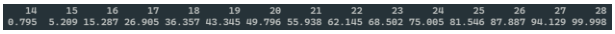

Figura 31. Tabla de frecuencias relativas acumuladas de edad.

Uniendo dichos resultados se arma la tabla de frecuencias, la cual es dada con el siguiente bloque de código.

```
tabla_frecs.edad <- cbind(vals.edad, fabsol.acum.edad, frelativas.edad, frelat.acum.edad)
colnames(tabla_frecs.edad) <- c("n", "N", "f", "F")
tabla_frecs.edad
```

Figura 32. Bloque de código que permite construir la tabla de frecuencias de edad.

Este bloque de código da como resultado la siguiente tabla.

4.1.2 TABLA DE FRECUENCIAS

	n	N	f	F
14	1446	1446	0.795	0.795
15	8029	9475	4.414	5.209
16	18332	27807	10.078	15.287
17	21134	48941	11.618	26.905
18	17194	66135	9.452	36.357
19	12711	78846	6.988	43.345
20	11734	90580	6.451	49.796
21	11173	101753	6.142	55.938
22	11291	113044	6.207	62.145
23	11564	124608	6.357	68.502
24	11829	136437	6.503	75.005
25	11899	148336	6.541	81.546
26	11535	159871	6.341	87.887
27	11355	171226	6.242	94.129
28	10676	181902	5.869	99.998

Figura 33. Tabla de frecuencias de edad.

4.1.3 TABLA DE DATOS AGRUPADOS

Para reunir los datos tal que se pueda dar como resultado la tabla de datos agrupados del conjunto de datos de edad, se almacenan datos como los que se verán a continuación en los códigos. Empezando por la cantidad de datos presentes en el conjunto, para esto se utiliza el siguiente código.

$$n.edad <- length(mydata.filtered.edad) \quad (14)$$

Este código da como resultado lo siguiente.

```
> n.edad
[1] 181902
```

Figura 34. Cantidad de datos en el conjunto de edad.

Con este dato se calcula la clase con la cual se darán a continuación los siguientes procesos, esto haciendo uso del siguiente código.

$$nclases.edad <- round(1 + log10(n.edad) / log10(2)) \quad (15)$$

Dicho código da como resultado lo siguiente.

```
> nclases.edad
[1] 18
```

Figura 35. Resultado de clases para edad.

Con lo dado es posible calcular el histograma con el siguiente código.

$$frecuencias.edad <- hist(mydata.filtered.edad, breaks = nclases.edad, right = FALSE, plot = F) \quad (16)$$

Código que da como resultado lo siguiente.

```
> frecuencias.edad
$breaks
 [1] 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

$counts
 [1]  1446  8029 18332 21134 17194 12711 11734 11173 11291 11564 11829 11899 11535 22031

$density
 [1] 0.007949335 0.044139152 0.100779541 0.116183439 0.094523425 0.069878286 0.064507262 0.061423184 0.062071885
[10] 0.063572693 0.065029521 0.065414344 0.063413266 0.121114666

$mids
 [1] 14.5 15.5 16.5 17.5 18.5 19.5 20.5 21.5 22.5 23.5 24.5 25.5 26.5 27.5

$xname
[1] "mydata.filtered.edad"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

Figura 36. Histograma de edad.

Este permite obtener las marcas de clase haciendo uso del siguiente código.

$$marcas\_clase.edad <- frecuencias.edad\$mids \quad (17)$$

Código el cual almacena el siguiente resultado.

```
[1] 14.5 15.5 16.5 17.5 18.5 19.5 20.5 21.5 22.5 23.5 24.5 25.5 26.5 27.5
```

Figura 37. Marcas de clase de edad.

El histograma también permite obtener las frecuencias absolutas, esto con el siguiente código.

$$fabsltas.edad <- frecuencias.edad\$counts \quad (18)$$

Este da como resultado la siguiente tabla.

```
[1]  1446  8029 18332 21134 17194 12711 11734 11173 11291 11564 11829 11899 11535 22031
```

Figura 38. Tabla de frecuencias absolutas de edad.

Del mismo modo este también permite obtener las frecuencias relativas, esto con el siguiente código.

$$freltvas.edad <- (round(fabsltas.edad / sum(fabsltas.edad), 3)) * 100 \quad (19)$$

Este código resulta en la siguiente tabla.

```
[1]  0.8  4.4 10.1 11.6  9.5  7.0  6.5  6.1  6.2  6.4  6.5  6.5  6.3 12.1
```

Figura 39. Tabla de frecuencias relativas de edad.

Continuando con las frecuencias acumuladas absolutas que se calculan con el siguiente código.

$$abs\_acum.edad \ <- cumsum(fabsltas.edad) \tag{20}$$

El cual da como resultado la siguiente tabla.

```
[1]   1446   9475  27807  48941  66135  78846  90580 101753 113044 124608 136437 148336 159871 181
```

Figura 40. Tabla de frecuencias absolutas acumuladas de edad.

Del mismo modo se calculan las frecuencias relativas acumuladas con el siguiente código.

$$rel\_acum.edad \ <- cumsum(freltvas.edad) \tag{21}$$

Este da como resultado la siguiente tabla.

```
[1]   0.8   5.2  15.3  26.9  36.4  43.4  49.9  56.0  62.2  68.6  75.1  81.6  87.9 100
```

Figura 41. Tabla de frecuencias relativas acumuladas de edad.

Haciendo uso de estos datos se forma la tabla deseada con el siguiente bloque de código.

```
tabla_dtsagr.edad <- cbind(marcas_clase.edad, fabsltas.edad, abs_acum.edad, freltvas.edad, rel_acum.edad)
colnames(tabla_dtsagr.edad) <- c("Mc", "n", "N", "f", "F")
rownames(tabla_dtsagr.edad) <- c("[14, 14.6)", "[14.6, 15.6)", "[15.6, 16.6)", "[16.6, 17.6)", "[17.6, 18.6
                                 "[18.6, 19.6)", "[19.6, 20.6)", "[20.6, 21.6)", "[21.6, 22.6)",
                                 "[22.6, 23.6)", "[23.6, 24.6)", "[24.6, 25.6)", "[25.6, 26.6)", "[26.6, 28
tabla_dtsagr.edad
```

Figura 42. Bloque de código que construye la tabla de datos agrupados de edad.

Dicho bloque de código da como resultado la siguiente tabla.

	Mc	n	N	f	F
[14, 14.6)	14.5	1446	1446	0.8	0.8
[14.6, 15.6)	15.5	8029	9475	4.4	5.2
[15.6, 16.6)	16.5	18332	27807	10.1	15.3
[16.6, 17.6)	17.5	21134	48941	11.6	26.9
[17.6, 18.6)	18.5	17194	66135	9.5	36.4
[18.6, 19.6)	19.5	12711	78846	7.0	43.4
[19.6, 20.6)	20.5	11734	90580	6.5	49.9
[20.6, 21.6)	21.5	11173	101753	6.1	56.0
[21.6, 22.6)	22.5	11291	113044	6.2	62.2
[22.6, 23.6)	23.5	11564	124608	6.4	68.6
[23.6, 24.6)	24.5	11829	136437	6.5	75.1
[24.6, 25.6)	25.5	11899	148336	6.5	81.6
[25.6, 26.6)	26.5	11535	159871	6.3	87.9
[26.6, 28]	27.5	22031	181902	12.1	100.0

Figura 43. Tabla de datos agrupados del conjunto de edad.

De estos datos también es posible graficar en un diagrama de barras el histograma de edades haciendo uso del siguiente código.

$$hist(mydata.filtered.edad, breaks = nclases.edad, right = FALSE, plot = T) \tag{22}$$

Del mismo resulta en el siguiente diagrama de barras.

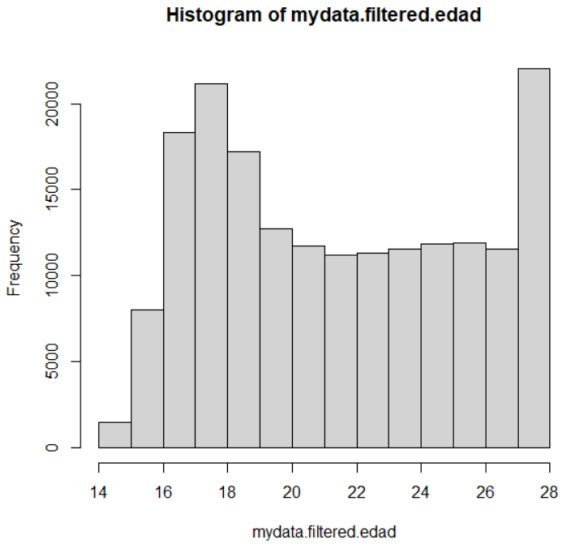

Figura 44. Diagrama de barras del histograma de las edades.

Luego se realiza el siguiente código con tal de obtener la sumatoria correspondiente.

$$sum(frecuencias.edad\$counts, 2) \tag{23}$$

El cual da el siguiente resultado.

```
[1] 181904
```

Figura 45. Resultado de la sumatoria.

Adicionalmente se puede comparar según Sturges haciendo uso del siguiente código.

$$frecuencias\_Comp.edad <- hist(mydata.filtered.edad, breaks = "Sturges", right = FALSE, plot = F) \tag{24}$$

El cual da como resultado el mismo histograma dado anteriormente, lo cual comprueba lo previamente hecho.

4.2 ESTRATO

Para el estrato, según los datos filtrados, se podrá obtener una mejor visión de la situación presente según el objetivo.

4.2.1 CALCULO PORCENTUAL

Se usa el siguiente bloque de código para reunir los valores del conjunto de datos e inicializar el porcentaje que acumulará cada dato.

```
perc.estrato = NULL
vals.estrato = table(mydata.filtered.estrato)
```

Figura 46. Acumulador de porcentaje y tabla del conjunto de datos.

Este bloque de código almacena cada cantidad de tipo de datos que contenga el conjunto de datos de estratos, tal como se puede ver a continuación.

```
mydata.filtered.estrato
    1     2     3
55431 76991 49480
```

Figura 47. Cantidad de cada tipo de dato que contiene el conjunto.

Posteriormente se almacenan los porcentajes correspondientes haciendo uso del siguiente código.

```
for (i in 1:length(vals.estrato)) {
  perc.estrato = round(cbind(perc.estrato, 100 * vals.estrato[[i]] / sum(vals.estrato)), 1)
}
```

Figura 48. Bloque de código que añade los porcentajes en el acumulador.

El acumulador toma la siguiente forma.

```
     [,1] [,2] [,3]
[1,] 30.5 42.3 27.2
```

Figura 49. Porcentajes acumulados.

A continuación se establecen las etiquetas que aparecerán en los diagramas en los cuales se podrán apreciar los datos correspondientes.

$$labels.estrato = paste("Estrato\ ", names(vals.estrato), "\ = \ ", perc.estrato, "\%") \quad (25)$$

Es entonces que se usa el siguiente código con el propósito de representar los datos mediante un diagrama circular.

$$pie(vals.estrato, labels = labels.estrato, main = \text{Estrato Aspirantes MINTIC 2022}) \quad (26)$$

Este código permite graficar de la siguiente forma.

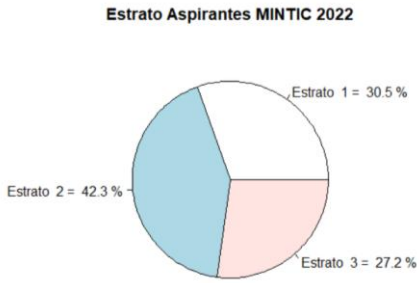

Figura 50. Diagrama circular de estratos con su respectivo porcentaje.

De estos datos se puede evidenciar que aquellos aspirantes con más presencia entre los jóvenes pertenecen a aquellos que se encuentran en un estrato 2. Esto se puede observar con también desde un diagrama de barras.

$$barplot(vals.estrato, main = "Estrato\ Aspirantes\ MINTIC\ 2022") \quad (27)$$

Este código permite graficar de la siguiente forma.

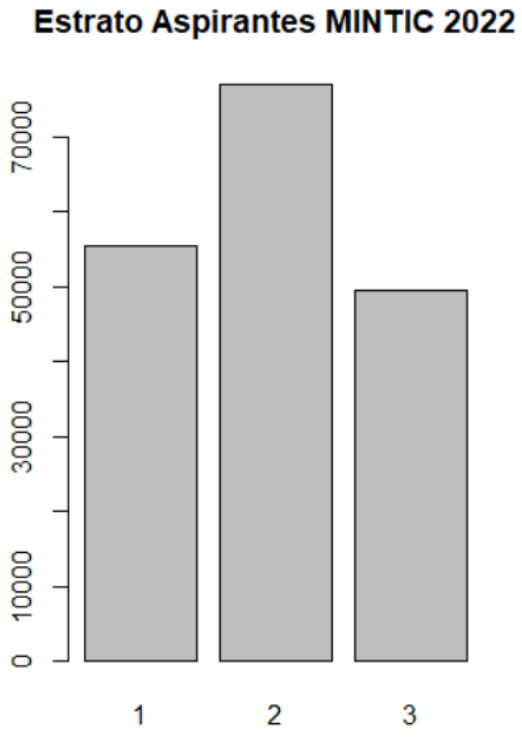

Figura 51. Diagrama de barras de estratos.

De estos datos también es posible sacar el diagrama de tallo y hojas que representa la distribución entre los datos, haciendo uso del siguiente código.

$$stem(mydata.filtered.estrato, scale = 0.2, width = 20, atom = 0.01) \quad (28)$$

Del cual resulta la siguiente tabla.

```
The decimal point is at the |

1 | 00000000+55411
2 | 00000000+76971
3 | 00000000+49460
```

Figura 52. Diagrama de tallo y hojas.

De este conjunto de datos se realiza el respectivo análisis estadístico haciendo uso del conjunto de las funciones dadas acumuladas en la función que será usada de la siguiente forma.

$$estdescript(mydata.filtered.estrato) \quad (29)$$

El resultado de este código es mostrado de la siguiente forma.

```
[1] "Media Aritmética: 1.96728458180779"
[1] "Mediana: 2"
[1] "Moda: 2"
[1] "Media Geométrica: Inf"
[1] "Media Armónica: 1.64736709437767"
[1] "Rango: 2"
[1] "Rango Intercuartílico: 2"
[1] "Varianza: 0.575677486909572"
[1] "Desviación Estándar: 0.758734134535657"
[1] "Coeficiente de Asimetría: 0.0545348599952335"
[1] "Curtosis: 1.73857005129823"
```

Figura 53. Resultados de análisis estadístico de estrato.

Además es posible sacar la normalización de este conjunto de datos, la cual es dada entre el mínimo y el máximo, esto haciendo uso del siguiente código.

```
print("Normalización entre el máximo y mínimo:")
xnormal.estrato = normalize(mydata.filtered.estrato)
```

Figura 54. Bloque de código de normalización de datos de estrato entre mínimo y máximo.

Este bloque de código da como resultado lo siguiente.

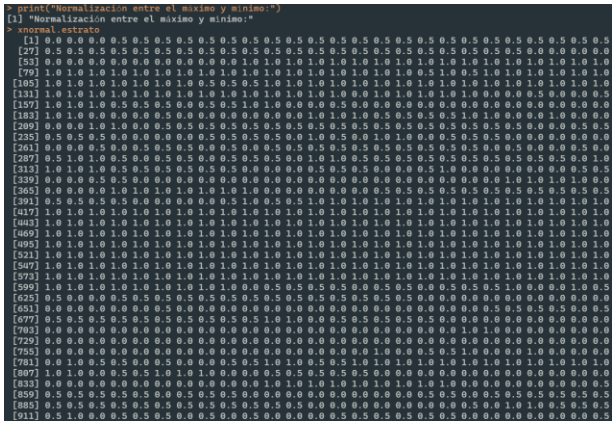

Figura 55. Parte de resultados de normalización de estrato entre máximo y mínimo.

De la misma manera también es posible la normalización con distribución normal con el siguiente código.

```
print("Normalización con la Distribución Normal:")
xnormz.estrato = znormalizar(mydata.filtered.estrato)
```

Figura 56. Bloque de código de normalización de datos de estrato con distribución normal.

Este bloque de código da como resultado lo siguiente.

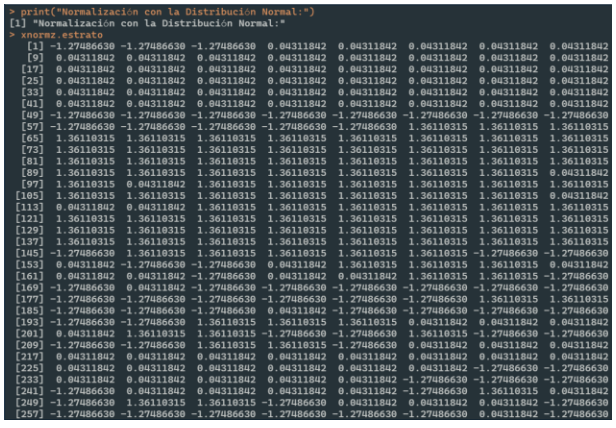

Figura 57. Parte de resultados de normalización de estrato con distribución normal.

4.2.2 TABLA DE FRECUENCIAS

Para reunir los datos tal que se pueda dar como resultado la tabla de frecuencia del conjunto de datos de estrato, se almacenan datos como los que se verán a continuación en los códigos. Empezando por las frecuencias relativas se utiliza el siguiente código.

$$frelativas.estrato <- round(prop.table(vals.estrato) * 100, 3) \quad (30)$$

Este da como resultado una tabla que almacena dichos datos, tal y como se ve a continuación.

```
mydata.filtered.estrato
    1     2     3
30.473 42.326 27.201
```

Figura 58. Tabla de frecuencias relativas de estrato.

Del mismo modo se obtiene la tabla de frecuencias absolutas acumuladas con el siguiente comando.

$$fabsol.acum.estrato <- cumsum(vals.estrato) \quad (31)$$

El cual da como resultado la siguiente tabla.

```
    1      2      3
55431 132422 181902
```

Figura 59. Tabla de frecuencias absolutas acumuladas de estrato.

De estos datos es posible sacar la frecuencia relativa acumulada con el siguiente código.

$$frelat.acum.estrato <- cumsum(frelativas.estrato) \quad (32)$$

El cual da como resultado la siguiente tabla.

```
    1      2      3
30.473  72.799 100.000
```

Figura 60. Tabla de frecuencias relativas acumuladas de estrato.

Uniendo dichos resultados se arma la tabla de frecuencias, la cual es dada con el siguiente bloque de código.

```
tabla_frecs.estrato <- cbind(vals.edad, fabsol.acum.estrato, frelativas.estrato, frelat.acum.estrato)
colnames(tabla_frecs.estrato) <- c("n", "N", "f", "F")
tabla_frecs.estrato
```

Figura 61. Bloque de código que permite construir la tabla de frecuencias de estrato.

Este bloque de código da como resultado la siguiente tabla.

	n	N	f	F
14	1446	55431	30.473	30.473
15	8029	132422	42.326	72.799
16	18332	181902	27.201	100.000
17	21134	55431	30.473	30.473
18	17194	132422	42.326	72.799
19	12711	181902	27.201	100.000
20	11734	55431	30.473	30.473
21	11173	132422	42.326	72.799
22	11291	181902	27.201	100.000
23	11564	55431	30.473	30.473
24	11829	132422	42.326	72.799
25	11899	181902	27.201	100.000
26	11535	55431	30.473	30.473
27	11355	132422	42.326	72.799
28	10676	181902	27.201	100.000

Figura 62. Tabla de frecuencias de estrato.

4.2.3 TABLA DE DATOS AGRUPADOS

Para reunir los datos tal que se pueda dar como resultado la tabla de datos agrupados del conjunto de datos de estrato, se almacenan datos como los que se verán a continuación en los códigos. Empezando por la cantidad de datos presentes en el conjunto, para esto se utiliza el siguiente código.

$$n.estrato <- length(mydata.filtered.estrato) \quad (33)$$

Este código da como resultado lo siguiente.

```
[1] 181902
```

Figura 63. Cantidad de datos en el conjunto de estrato.

Con este dato se calcula la clase con la cual se darán a continuación los siguientes procesos, esto haciendo uso del siguiente código.

$$nclases.estrato <- round(1 + log10(n.estrato) / log10(2)) \quad (34)$$

Dicho código da como resultado lo siguiente.

```
[1] 18
```

Figura 64. Resultado de clases para estrato.

Con lo dado es posible calcular el histograma con el siguiente código.

$$frecuencias.estrato <- hist(mydata.filtered.estrato, breaks = nclases.estrato, right = FALSE, plot = F) \quad (35)$$

Código que da como resultado lo siguiente.

```
$breaks
 [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0

$counts
 [1] 55431     0     0     0     0     0     0     0     0 76991     0     0     0     0     0     0     0
[19]     0 49480

$density
 [1] 3.047300 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 4.232554 0.000000
[13] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 2.720146

$mids
 [1] 1.05 1.15 1.25 1.35 1.45 1.55 1.65 1.75 1.85 1.95 2.05 2.15 2.25 2.35 2.45 2.55 2.65 2.75 2.85 2.95

$xname
[1] "mydata.filtered.estrato"

$equidist
[1] TRUE

attr(,"class")
[1] "histogram"
```

Figura 65. Histograma de estrato.

Este permite obtener las marcas de clase haciendo uso del siguiente código.

$$marcas\_clase.estrato <- frecuencias.estrato\$mids \quad (36)$$

Código el cual almacena el siguiente resultado.

```
[1] 1.05 1.15 1.25 1.35 1.45 1.55 1.65 1.75 1.85 1.95 2.05 2.15 2.25 2.35 2.45 2.55 2.65 2.75 2.85 2.95
```

Figura 66. Marcas de clase de estrato.

El histograma también permite obtener las frecuencias absolutas, esto con el siguiente código.

$$fabsltas.estrato <- frecuencias.estrato\$counts \quad (37)$$

Este da como resultado la siguiente tabla.

```
 [1] 55431     0     0     0     0     0     0     0     0 76991     0     0     0     0     0     0     0
[19]     0 49480
```

Figura 67. Tabla de frecuencias absolutas de estrato.

Del mismo modo este también permite obtener las frecuencias relativas, esto con el siguiente código.

$$freltvas.estrato <- (round(fabsltas.estrato / sum(fabsltas.estrato), 3)) * 100 \quad (38)$$

Este código resulta en la siguiente tabla.

```
[1] 30.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 42.3  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 27.2
```

Figura 68. Tabla de frecuencias relativas de estrato.

Continuando con las frecuencias acumuladas absolutas que se calculan con el siguiente código.

$$abs\_acum.estrato <- cumsum(fabsltas.estrato) \quad (39)$$

El cual da como resultado la siguiente tabla.

```
 [1]  55431  55431  55431  55431  55431  55431  55431  55431  55431  55431 132422 132422 132422 132422 132422
[16] 132422 132422 132422 132422 132422 181902
```

Figura 69. Tabla de frecuencias absolutas acumuladas de estrato.

Del mismo modo se calculan las frecuencias relativas acumuladas con el siguiente código.

$$rel\_acum.estrato <- cumsum(freltvas.estrato) \quad (40)$$

Este da como resultado la siguiente tabla.

```
 [1]  30.5  30.5  30.5  30.5  30.5  30.5  30.5  30.5  30.5  30.5  72.8  72.8  72.8  72.8  72.8  72.8  72.8  72.8
[19]  72.8 100.0
```

Figura 70. Tabla de frecuencias relativas acumuladas de estrato.

Haciendo uso de estos datos se forma la tabla deseada con el siguiente bloque de código.

```
tabla_dtsagr.estrato <- cbind(marcas_clase.estrato, fabsltas.estrato,
                              abs_acum.estrato, freltvas.estrato, rel_acum.estrato)
colnames(tabla_dtsagr.estrato) <- c("Mc", "n", "N", "f", "F")
rownames(tabla_dtsagr.estrato) <- c("[1, 1.06)", "[1.06, 1.16)", "[1.16, 1.26)", "[1.26, 1.36)",
                                    "[1.36, 1.46)", "[1.46, 1.56)", "[1.56, 1.66)", "[1.66, 1.76)",
                                    "[1.76, 1.86)", "[1.86, 1.96)", "[1.96, 2.06)", "[2.06, 2.16)",
                                    "[2.16, 2.26)", "[2.26, 2.36)", "[2.36, 2.46)", "[2.46, 2.56)",
                                    "[2.56, 2.66)", "[2.66, 2.76)", "[2.76, 2.86)", "[2.86, 3]")
tabla_dtsagr.estrato
```

Figura 71. Bloque de código que construye la tabla de datos agrupados de estrato.

Dicho bloque de código da como resultado la siguiente tabla.

	Mc	n	N	f	F
[1, 1.06)	1.05	55431	55431	30.5	30.5
[1.06, 1.16)	1.15	0	55431	0.0	30.5
[1.16, 1.26)	1.25	0	55431	0.0	30.5
[1.26, 1.36)	1.35	0	55431	0.0	30.5
[1.36, 1.46)	1.45	0	55431	0.0	30.5
[1.46, 1.56)	1.55	0	55431	0.0	30.5
[1.56, 1.66)	1.65	0	55431	0.0	30.5
[1.66, 1.76)	1.75	0	55431	0.0	30.5
[1.76, 1.86)	1.85	0	55431	0.0	30.5
[1.86, 1.96)	1.95	0	55431	0.0	30.5
[1.96, 2.06)	2.05	76991	132422	42.3	72.8
[2.06, 2.16)	2.15	0	132422	0.0	72.8
[2.16, 2.26)	2.25	0	132422	0.0	72.8
[2.26, 2.36)	2.35	0	132422	0.0	72.8
[2.36, 2.46)	2.45	0	132422	0.0	72.8
[2.46, 2.56)	2.55	0	132422	0.0	72.8
[2.56, 2.66)	2.65	0	132422	0.0	72.8
[2.66, 2.76)	2.75	0	132422	0.0	72.8
[2.76, 2.86)	2.85	0	132422	0.0	72.8
[2.86, 3]	2.95	49480	181902	27.2	100.0

Figura 72. Tabla de datos agrupados del conjunto de estrato.

De estos datos también es posible graficar en un diagrama de barras el histograma de estratos haciendo uso del siguiente código.

$$hist(mydata.filtered.estrato, breaks = nclases.estrato, right = FALSE, plot = T) \quad (41)$$

Del mismo resulta en el siguiente diagrama de barras.

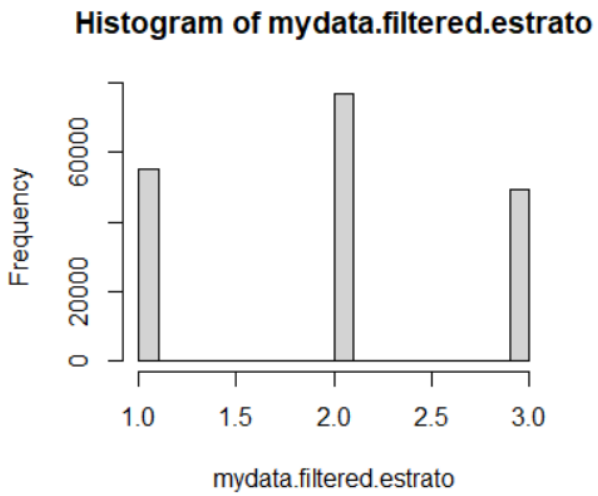

Figura 73. Diagrama de barras del histograma de los estratos.

Luego se realiza el siguiente código con tal de obtener la sumatoria correspondiente.

$$sum(frecuencias.estrato\$counts, 2) \quad (42)$$

El cual da el siguiente resultado.

```
[1] 181904
```

Figura 74. Resultado de la sumatoria.

Adicionalmente se puede comparar según Sturges haciendo uso del siguiente código.

$$frecuencias\_Comp.estrato <- hist(mydata.filtered.estrato, breaks = "Sturges", right = FALSE, plot = F) \quad (43)$$

El cual da como resultado el mismo histograma dado anteriormente, lo cual comprueba lo previamente hecho.

4.3 UBICACIÓN

Para la ubicación, según los datos filtrados, se podrá obtener una mejor visión de la situación presente según el objetivo, estos siendo interpretados tal que sea posible analizarlos así sus valores sean cualitativos.

4.3.1 CALCULO PORCENTUAL

Se usa el siguiente bloque de código para reunir los valores del conjunto de datos e inicializar el porcentaje que acumulará cada dato.

```
perc.ubicacion = NULL
vals.ubicacion = table(mydata.filtered.ubicacion)
```

Figura 75. Acumulador de porcentaje y tabla del conjunto de datos.

Este bloque de código almacena cada cantidad de tipo de datos que contenga el conjunto de datos de ubicaciones, tal como se puede ver a continuación.

```
          ABRIAQUI, ANTIOQUIA                    ACACIAS, META
                            3                              664
               ACANDI, CHOCO                  ACEVEDO, HUILA
                           13                              119
               ACHI, BOLIVAR                  AGRADO, HUILA
                           61                               62
  AGUA DE DIOS, CUNDINAMARCA                AGUACHICA, CESAR
                          232                              413
            AGUADA, SANTANDER                AGUADAS, CALDAS
                            5                              190
            AGUAZUL, CASANARE            AGUSTIN CODAZZI, CESAR
                          346                              252
               AIPE, HUILA                  ALBAN, CUNDINAMARCA
                          135                               63
              ALBAN, NARIÑO                  ALBANIA, CAQUETA
                           42                               10
          ALBANIA, LA GUAJIRA                ALBANIA, SANTANDER
                           75                               29
      ALCALA, VALLE DEL CAUCA                  ALDANA, NARIÑO
                           81                               26
        ALEJANDRIA, ANTIOQUIA                ALGARROBO, MAGDALENA
                           25                               45
            ALGECIRAS, HUILA                  ALMAGUER, CAUCA
                           98                               15
             ALMEIDA, BOYACA                ALPUJARRA, TOLIMA
                           10                               30
             ALTAMIRA, HUILA                ALTO BAUDO, CHOCO
                           30                               10
    ALTOS DEL ROSARIO, BOLIVAR                ALVARADO, TOLIMA
                           22                               37
              AMAGA, ANTIOQUIA                  AMALFI, ANTIOQUIA
                           87                               50
            AMBALEMA, TOLIMA                ANAPOIMA, CUNDINAMARCA
                           35                              128
             ANCUYA, NARIÑO                ANDALUCIA, VALLE DEL CAUCA
                           32                              112
              ANDES, ANTIOQUIA                ANGELOPOLIS, ANTIOQUIA
                          123                               17
```

Figura 76. Cantidad de cada tipo de dato que contiene el conjunto.

Posteriormente se almacenan los porcentajes correspondientes haciendo uso del siguiente código.

```
for (i in 1:length(vals.ubicacion)) {
  perc.ubicacion = round(cbind(perc.ubicacion, 100 * vals.ubicacion[[i]] / sum(vals.ubicacion)), 1)
}
```

Figura 77. Bloque de código que añade los porcentajes en el acumulador.

El acumulador toma la siguiente forma.

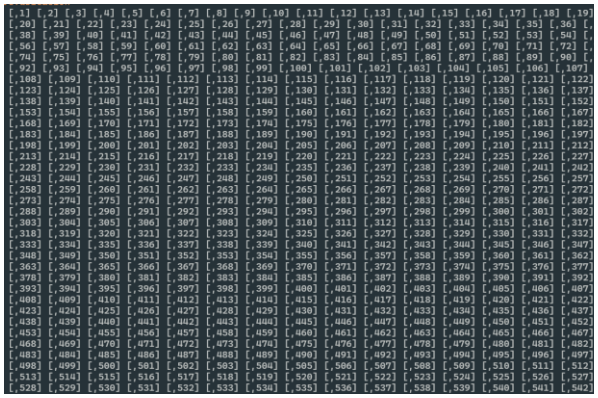

Figura 78. Parte de porcentajes acumulados.

A continuación se establecen las etiquetas que aparecerán en los diagramas en los cuales se podrán apreciar los datos correspondientes.

$$labels.ubicacion = paste(names(vals.ubicacion), " = ", perc.ubicacion, "\%") \quad (44)$$

Es entonces que se usa el siguiente código con el propósito de representar los datos mediante un diagrama circular.

$$pie(vals.ubicacion, labels = labels.ubicacion, main = "Ubicaciones\ Aspirantes\ MINTIC\ 2022", radius = 1.2, cex = 0.6) \quad (45)$$

Este diagrama se verá de la siguiente forma.

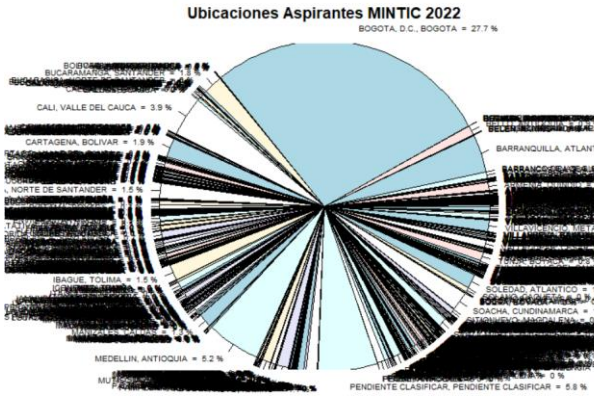

Figura 79. Diagrama circular de ubicaciones específicas con su respectivo porcentaje.

Como se puede ver en la figura 79, la inmensa cantidad de variación entre ubicaciones es tan alta que las etiquetas del diagrama se superponen, por esta razón también se grafica en un diagrama de barras con el siguiente código.

```
par(mar = c(4,15,1,1)) # BLTR
barplot(vals.ubicacion, main = "Ubicaciones Aspirantes MINTIC 2022", horiz = T,
        las = 2, names.arg = labels.ubicacion, width = 0.2)
```

Figura 80. Bloque de código personalizado para construir diagrama de barras.

Este código permitirá graficar de forma que no se ofuscan demasiados datos.

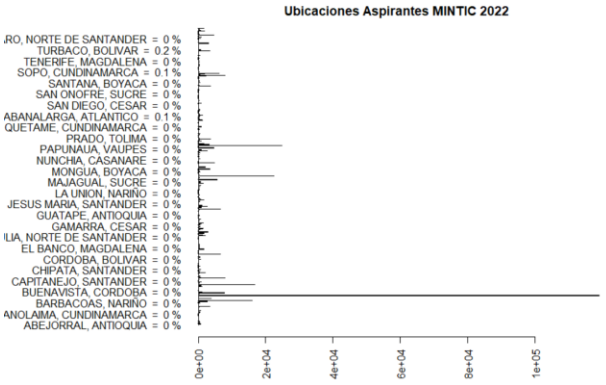

Figura 81. Diagrama de barras de ubicaciones específicas.

Sin embargo, como se observa en la figura 81 se sigue ocultando información, por cómo se observa que hay cantidades muy pequeñas porcentuales de ubicaciones. Tal que tomando desde el diagrama circular podemos decir que la ubicación con mayor porcentaje de población que cumple con los requerimientos se encuentra en Bogotá D.C, Bogotá con un 27,7%.

Por esta razón se decidió generalizar las ubicaciones, esto basándose en limitar el alcance al departamento.

Esto se puede lograr implementando el siguiente código en el que se retoma lo previo pero se cambia el conjunto de datos.

Empezando con este bloque de código.

```
perc.ubicacion.dep = NULL
vals.ubicacion.dep = table(mydata.filtered.ubicacion.general)
```

Figura 82. Acumulador de porcentaje y tabla del conjunto de datos.

Este bloque de código almacena cada cantidad de tipo de datos que contenga el conjunto de datos de ubicaciones, tal como se puede ver a continuación.

```
mydata.filtered.ubicacion.general
      AMAZONAS          ANTIOQUIA             ARAUCA           ATLANTI
           212              44001               1353              261
       BOGOTA            BOLIVAR             BOYACA             CALDA
        19093              11349              12724              112
      CAQUETA          CASANARE             CAUCA              CES
         2297               3215               6282              57
        CHOCO            CORDOBA       CUNDINAMARCA            GUAIN
         1284               7120              30270               1
     GUAVIARE            HUILA        LA GUAJIRA          MAGDALE
          284               9364               2858               60
         META           NARIÑO    NORTE DE SANTANDER     PENDIENTE CLASIFIC
         6845               8516              10088              260
     PUTUMAYO          QUINDIO     RISARALDA SAN ANDRES Y PROVIDENC
         1424               4995               5369                2
     SANTANDER           SUCRE             TOLIMA          VALLE DEL CAU
        21633              4144              10635              287
       VAUPES         VICHADA
          167             190
```

Figura 82. Cantidad de cada tipo de dato que contiene el conjunto.

Posteriormente se almacenan los porcentajes correspondientes haciendo uso del siguiente código.

```
for (i in 1:length(vals.ubicacion.dep)) {
  perc.ubicacion.dep = round(cbind(perc.ubicacion.dep, 100 * vals.ubicacion.dep[[i]] / sum(vals.ubicacion.dep)), 1)
}
```

Figura 83. Bloque de código que añade los porcentajes en el acumulador.

El acumulador toma la siguiente forma.

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
[1,]  0.3 10.2  0.3  6.1 27.7  2.6    3  2.6  0.5   0.7   1.5   1.3   0.3   1.7     7     0   0.1   2.2   0.7   1.4
     [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38] [,39]
[1,]   1.6     2   2.5   5.8   0.3   1.2   1.2   0.1     5     1   2.5   6.7     0     0     0  10.2   0.3   6.1  27.7
     [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50] [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58]
[1,]   2.6     3   2.6   0.5   0.7   1.5   1.3   0.3   1.7     7     0   0.1   2.2   0.7   1.4   1.6     2   2.5   5.8
     [,59] [,60] [,61] [,62] [,63] [,64] [,65] [,66] [,67] [,68]
[1,]  0.3   1.2   1.2   0.1     5     1   2.5   6.7     0     0
```

Figura 84. Parte de porcentajes acumulados.

A continuación se establecen las etiquetas que aparecerán en los diagramas en los cuales se podrán apreciar los datos correspondientes.

$$labels.ubicacion.dep = paste(names(vals.ubicacion.dep), " = ", perc.ubicacion.dep, "\%") \quad (46)$$

Es entonces que se usa el siguiente código con el propósito de representar los datos mediante un diagrama circular.

$$pie(vals.ubicacion.dep, labels = labels.ubicacion.dep, main = \text{Ubicaciones (Departamentos) Aspirantes MINTIC 2022}, radius = 1.2, cex = 0.6) \quad (47)$$

Este código permite graficar de la siguiente forma.

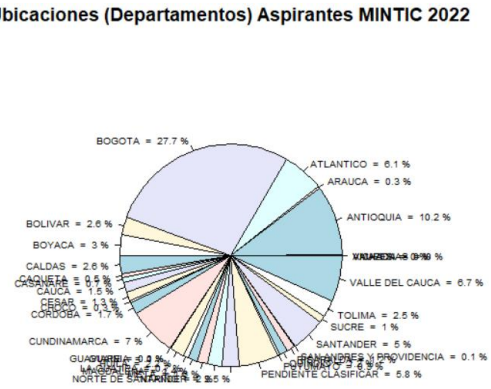

Figura 85. Diagrama circular de ubicaciones generales con su respectivo porcentaje.

Como se puede ver en la figura 85, la inmensa cantidad de variación entre ubicaciones específicas se ha vuelto general y aun es tan alta que algunas etiquetas del diagrama siguen superponiéndose, aunque es más legible, por esta razón también se grafica en un diagrama de barras con el siguiente código.

```
par(mar = c(4,15,1,1)) # BLTR
barplot(vals.ubicacion.dep, main = "Ubicaciones (Departamentos) Aspirantes MINTIC 2022",
        horiz = T, las = 2, names.arg = labels.ubicacion.dep, width = 0.2)
```

Figura 86. Bloque de código personalizado para construir diagrama de barras.

Este código permitirá graficar de forma que no se ofuscan demasiados datos.

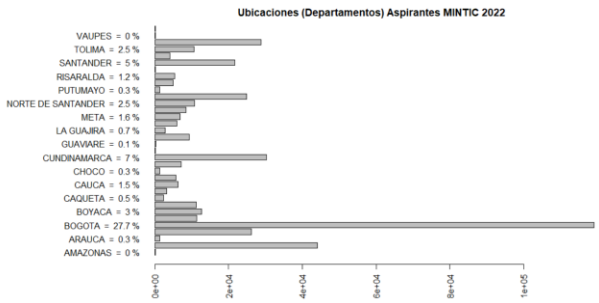

Figura 87. Diagrama de barras de ubicaciones generales.

Tal como se ve en la figura 87, el registro de ubicaciones se entiende con mayor facilidad, así como se puede ver que hay datos tan pequeños que son redondeados a 0, a la vez que existen algunos tan grandes que salen de la gráfica.

5 CONCLUSIÓN

Para terminar, se puede decir que con los datos analizados las personas jóvenes de entre 14 a 28 años que pertenecen a los estratos medios o bajos se ubican en su mayoría en Bogotá D.C., en la cual el estrato más común es el 2 y la edad de quienes más ingresan al programa son chicos y chicas de 17 años. Con esto se puede confirmar que quienes más buscan tener puertas abiertas para el conocimiento, sobre todo en esta época de tecnología, son aquellos jóvenes que pasaran a su mayoría de edad y su estrato los mantiene limitados tal que tengan que aprovechar cualquier oportunidad para asegurar su futuro. Así como también vale la pena resaltar la facilidad y manejabilidad de los datos haciendo uso del lenguaje de programación R y su entorno de desarrollo integrado RStudio, que con ayuda de los plugins necesarios se llevó a cabo el proyecto con mejor rendimiento y efectividad.

6 REFERENCIAS

[1] Ministerio de Tecnologías de La Información y Las Comunicaciones, “Misión TIC 2020 100 mil programadores”, Misión TIC 2020 100 mil programadores. 15-ene-2021.

Adaptado por:
PunkerGhoul, para el curso de Probabilidad y Estadística.
2021