

# Presentacion\_ProyectoFinal

PunkerGhoul

30/10/2021

## MINTIC 2022 - Análisis

### Preparación

```
rm(list=ls())
```

### Librerías

```
library(moments)
```

```
## Warning: package 'moments' was built under R version 4.1.1
```

### Funciones que se usarán para los procedimientos

#### Medidas de tendencia central

```
suma = function(x, y) {  
  return(x + y)  
}  
getmode <- function(v) {  
  uniqv <- unique(v)  
  uniqv[which.max(tabulate(match(v, uniqv)))]  
}  
normalize <- function(x) {  
  return((x - min(x)) / (max(x) - min(x)))  
}  
znormalizar <- function(x) {  
  return((x - mean(x)) / sd(x))  
}  
mgeometrica <- function(x) {  
  n = length(x)  
  producto = prod(x)  
  return(producto ** (1 / n))  
}
```

```

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

```

## Medidas de variación

```

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

```

## Estadística descriptiva

```

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)))
}

```

## Importar DataSet

```

mydata <- read.csv2(file = "Datasets/Misi_n_TIC_2020_100_mil_programadores_CSV.csv",
  sep = ",", encoding = "UTF-8")
#View(mydata)
names(mydata)

```

```

## [1] "FECHA_CORTE"      "DEPARTAME_NOMBRE" "MUNICIPIO_NOMBRE"
## [4] "ESTADO_FORMACION" "ESTADOS"          "CONVOCATORIA"
## [7] "ESTRATO_SOCIAL"   "EDAD"             "GENERO"
## [10] "RUTA"            "GRADO"            "META_BENEFICIARIO"
## [13] "INVERSION"       "META_DINERO"      "FECHA_VIGENCIA"

```

## Exploración del DataSet y gráficos

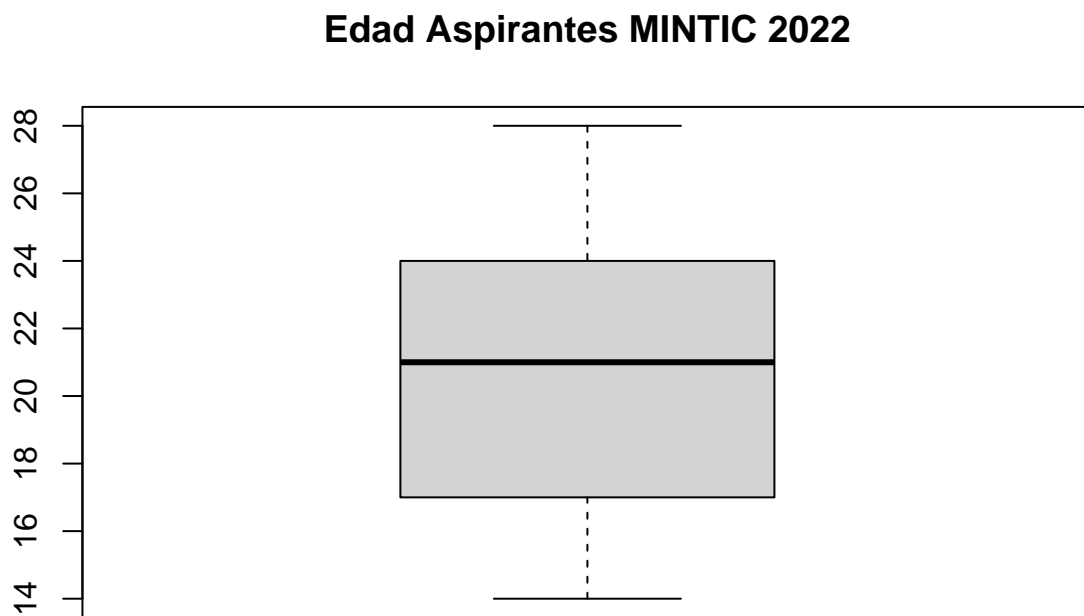
Filtrar DataSet según requerimientos, Jóvenes de estrato bajo o medio

```
mydata.filtered <- subset(mydata, (mydata$EDAD > 13 & mydata$EDAD < 29) &
                             (mydata$ESTRATO_SOCIAL < 4))
mydata.filtered.edad <- mydata.filtered$EDAD
mydata.filtered.estrato <- as.integer(mydata.filtered$ESTRATO_SOCIAL)

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

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

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



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

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

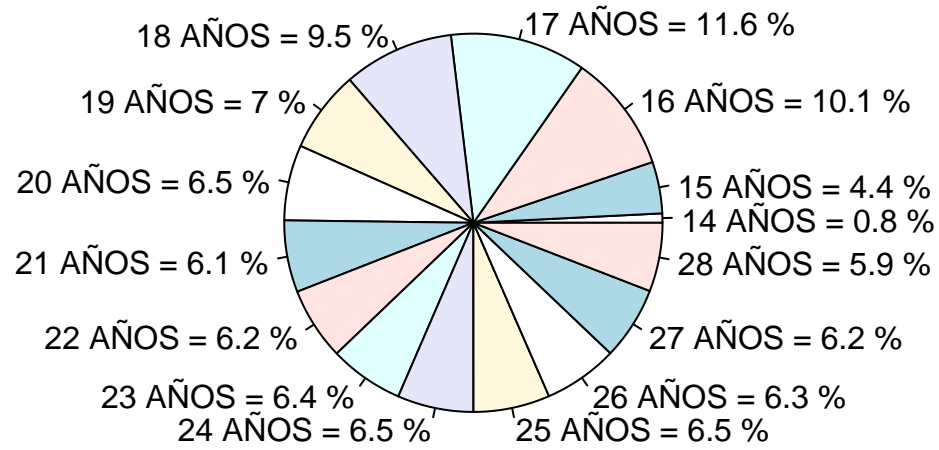
```
boxplot(mydata.filtered.estrato, main = "Estrato Aspirantes MINTIC 2022")
```



#### Cálculo, gráficos y diagramas - EDAD

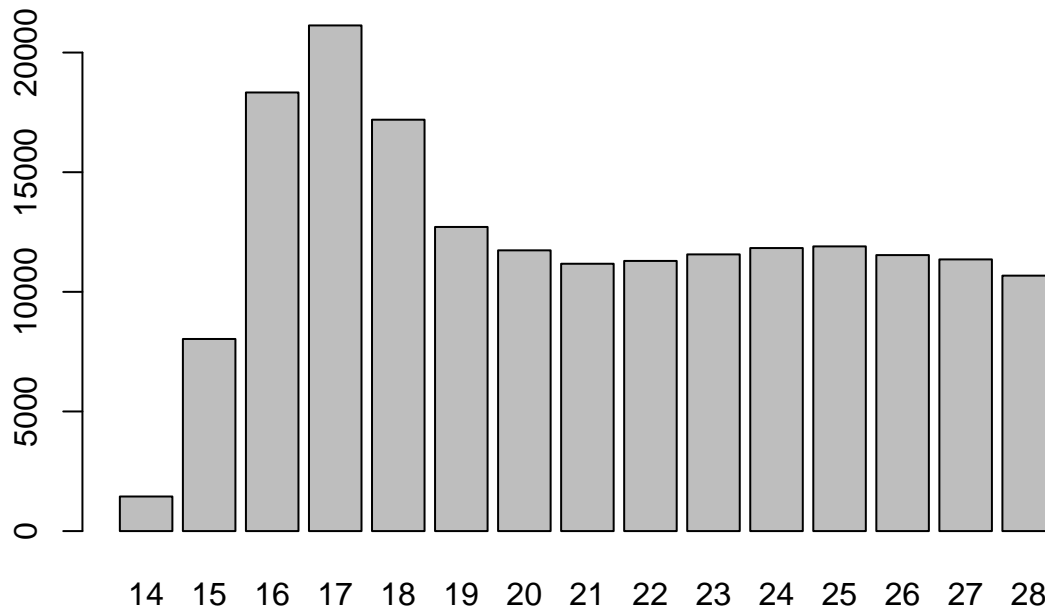
```
perc.edad = NULL
vals.edad = table(mydata.filtered.edad)
for (i in 1:length(vals.edad)) {
  perc.edad = round(cbind(perc.edad, 100 * vals.edad[[i]] / sum(vals.edad)), 1)
}
labels.edad = paste(names(vals.edad), "AÑOS =", perc.edad, "%")
pie(vals.edad, labels = labels.edad, main = "Edad Aspirantes MINTIC 2022")
```

## Edad Aspirantes MINTIC 2022



```
barplot(vals.edad, main = "Edad Aspirantes MINTIC 2022")
```

## Edad Aspirantes MINTIC 2022



```
stem(mydata.filtered.edad, scale = 1, width = 20, atom = 0.01)
```

```
##
## 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
```

```
estdescript(mydata.filtered.edad)
```

```
## [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"
```

```
print("Normalización entre el máximo y mínimo:")
```

```
## [1] "Normalización entre el máximo y mínimo:"
```

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

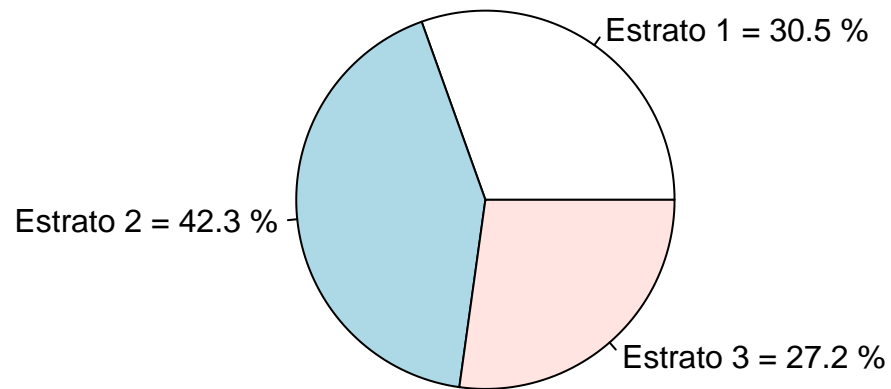
```
## [1] "Normalización con la Distribución Normal:"
```

```
xnormz.edad = znormalizar(mydata.filtered.edad)
```

## Cálculo, gráficos y diagramas - ESTRATO

```
perc.estrato = NULL
vals.estrato = table(mydata.filtered.estrato)
for (i in 1:length(vals.estrato)) {
  perc.estrato = round(cbind(perc.estrato,
                             100 * vals.estrato[[i]] / sum(vals.estrato)), 1)
}
labels.estrato = paste("Estrato", names(vals.estrato), "=", perc.estrato, "%")
pie(vals.estrato, labels = labels.estrato,
    main = "Estrato Aspirantes MINTIC 2022")
```

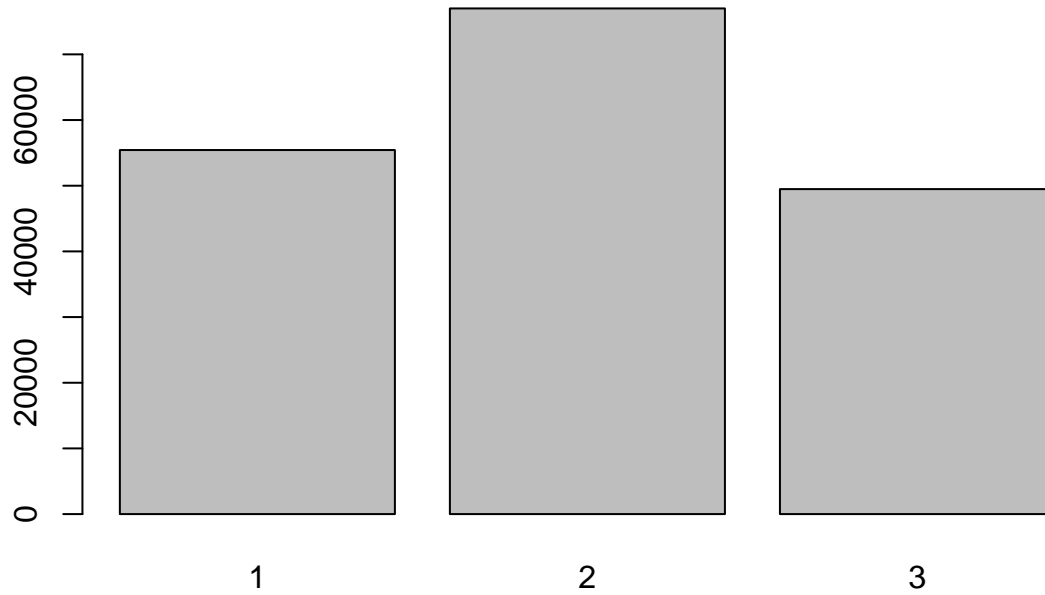
## Estrato Aspirantes MINTIC 2022



```
barplot(vals.estrato, main = "Estrato Aspirantes MINTIC 2022")
```



## Estrato Aspirantes MINTIC 2022



```
stem(mydata.filtered.estrato, scale = 1, width = 20, atom = 0.01)
```

```
##
## The decimal point is 1 digit(s) to the left of the |
##
## 10 | 00000000+55411
## 11 |
## 12 |
## 13 |
## 14 |
## 15 |
## 16 |
## 17 |
## 18 |
## 19 |
## 20 | 00000000+76971
## 21 |
## 22 |
## 23 |
## 24 |
## 25 |
## 26 |
## 27 |
## 28 |
## 29 |
## 30 | 00000000+49460
```

```
estdescript(mydata.filtered.estrato)
```

```
## [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"
```

```
print("Normalización entre el máximo y mínimo:")
```

```
## [1] "Normalización entre el máximo y mínimo:"
```

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

```
## [1] "Normalización con la Distribución Normal:"
```

```
xnormz.estrato = znormalizar(mydata.filtered.estrato)
```

## Tabla de Frecuencias - EDAD

```
frelativas.edad <- round(prop.table(vals.edad) * 100, 3)
fabsol.acum.edad <- cumsum(vals.edad)
frelat.acum.edad <- cumsum(frelativas.edad)
tabla.edad <- cbind(vals.edad, fabsol.acum.edad, frelativas.edad,
                    frelat.acum.edad)
colnames(tabla.edad) <- c("n", "N", "f", "F")
tabla.edad
```

```
##      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
```

## Tabla de Frecuencias - ESTRATO

```
frelativas.estrato <- round(prop.table(vals.estrato) * 100, 3)
fabsol.acum.estrato <- cumsum(vals.estrato)
frelat.acum.estrato <- cumsum(frelativas.estrato)
tabla.estrato <- cbind(vals.edad, fabsol.acum.estrato, frelativas.estrato,
                      frelat.acum.estrato)
colnames(tabla.estrato) <- c("n", "N", "f", "F")
tabla.estrato
```

| ##    | 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 |