Laboratorio3

Cristian

2025-08-13

```
library(ggplot2)
library(GGally)
library(dplyr)
library(RColorBrewer)
library(gtools)
library(modeest)
library(psych)
```

Permutaciones

```
personas <- c("Ana", "Luis", "Marta", "Carlos")
formas_de_ordenar <- permutations(length(personas), length(personas),
personas)
cat("Formas posibles de ordenar las llaves mágicas:\n")
```

```
## Formas posibles de ordenar las llaves mágicas:
```

```
print(formas_de_ordenar)
```

```
##       [,1]     [,2]     [,3]     [,4]
##  [1,] "Ana"    "Carlos" "Luis"   "Marta"
##  [2,] "Ana"    "Carlos" "Marta"  "Luis"
##  [3,] "Ana"    "Luis"   "Carlos" "Marta"
##  [4,] "Ana"    "Luis"   "Marta"  "Carlos"
##  [5,] "Ana"    "Marta"  "Carlos" "Luis"
##  [6,] "Ana"    "Marta"  "Luis"   "Carlos"
##  [7,] "Carlos" "Ana"    "Luis"   "Marta"
##  [8,] "Carlos" "Ana"    "Marta"  "Luis"
##  [9,] "Carlos" "Luis"   "Ana"    "Marta"
## [10,] "Carlos" "Luis"   "Marta"  "Ana"
## [11,] "Carlos" "Marta"  "Ana"    "Luis"
## [12,] "Carlos" "Marta"  "Luis"   "Ana"
## [13,] "Luis"   "Ana"    "Carlos" "Marta"
## [14,] "Luis"   "Ana"    "Marta"  "Carlos"
## [15,] "Luis"   "Carlos" "Ana"    "Marta"
## [16,] "Luis"   "Carlos" "Marta"  "Ana"
## [17,] "Luis"   "Marta"  "Ana"    "Carlos"
## [18,] "Luis"   "Marta"  "Carlos" "Ana"
## [19,] "Marta"  "Ana"    "Carlos" "Luis"
## [20,] "Marta"  "Ana"    "Luis"   "Carlos"
## [21,] "Marta"  "Carlos" "Ana"    "Luis"
## [22,] "Marta"  "Carlos" "Luis"   "Ana"
```

```
## [23,] "Marta"  "Luis"   "Ana"    "Carlos"
## [24,] "Marta"  "Luis"   "Carlos" "Ana"

total_permutaciones <- factorial(length(personas))
cat("Total de formas posibles:", total_permutaciones, "\n")

## Total de formas posibles: 24
```

Combinaciones

```
coleccion_libros <- 1:10
grupos_posibles <- combinations(length(coleccion_libros), 3,
coleccion_libros)

cat("Grupos distintos que puede formar Ana con 3 libros:\n")

## Grupos distintos que puede formar Ana con 3 libros:

print(grupos_posibles)

##        [,1] [,2] [,3]
##   [1,]    1    2    3
##   [2,]    1    2    4
##   [3,]    1    2    5
##   [4,]    1    2    6
##   [5,]    1    2    7
##   [6,]    1    2    8
##   [7,]    1    2    9
##   [8,]    1    2   10
##   [9,]    1    3    4
##  [10,]    1    3    5
##  [11,]    1    3    6
##  [12,]    1    3    7
##  [13,]    1    3    8
##  [14,]    1    3    9
##  [15,]    1    3   10
##  [16,]    1    4    5
##  [17,]    1    4    6
##  [18,]    1    4    7
##  [19,]    1    4    8
##  [20,]    1    4    9
##  [21,]    1    4   10
##  [22,]    1    5    6
##  [23,]    1    5    7
##  [24,]    1    5    8
##  [25,]    1    5    9
##  [26,]    1    5   10
##  [27,]    1    6    7
##  [28,]    1    6    8
##  [29,]    1    6    9
```

```
##  [30,]    1    6   10
##  [31,]    1    7    8
##  [32,]    1    7    9
##  [33,]    1    7   10
##  [34,]    1    8    9
##  [35,]    1    8   10
##  [36,]    1    9   10
##  [37,]    2    3    4
##  [38,]    2    3    5
##  [39,]    2    3    6
##  [40,]    2    3    7
##  [41,]    2    3    8
##  [42,]    2    3    9
##  [43,]    2    3   10
##  [44,]    2    4    5
##  [45,]    2    4    6
##  [46,]    2    4    7
##  [47,]    2    4    8
##  [48,]    2    4    9
##  [49,]    2    4   10
##  [50,]    2    5    6
##  [51,]    2    5    7
##  [52,]    2    5    8
##  [53,]    2    5    9
##  [54,]    2    5   10
##  [55,]    2    6    7
##  [56,]    2    6    8
##  [57,]    2    6    9
##  [58,]    2    6   10
##  [59,]    2    7    8
##  [60,]    2    7    9
##  [61,]    2    7   10
##  [62,]    2    8    9
##  [63,]    2    8   10
##  [64,]    2    9   10
##  [65,]    3    4    5
##  [66,]    3    4    6
##  [67,]    3    4    7
##  [68,]    3    4    8
##  [69,]    3    4    9
##  [70,]    3    4   10
##  [71,]    3    5    6
##  [72,]    3    5    7
##  [73,]    3    5    8
##  [74,]    3    5    9
##  [75,]    3    5   10
##  [76,]    3    6    7
##  [77,]    3    6    8
##  [78,]    3    6    9
##  [79,]    3    6   10
```

```
##  [80,]    3    7    8
##  [81,]    3    7    9
##  [82,]    3    7   10
##  [83,]    3    8    9
##  [84,]    3    8   10
##  [85,]    3    9   10
##  [86,]    4    5    6
##  [87,]    4    5    7
##  [88,]    4    5    8
##  [89,]    4    5    9
##  [90,]    4    5   10
##  [91,]    4    6    7
##  [92,]    4    6    8
##  [93,]    4    6    9
##  [94,]    4    6   10
##  [95,]    4    7    8
##  [96,]    4    7    9
##  [97,]    4    7   10
##  [98,]    4    8    9
##  [99,]    4    8   10
## [100,]    4    9   10
## [101,]    5    6    7
## [102,]    5    6    8
## [103,]    5    6    9
## [104,]    5    6   10
## [105,]    5    7    8
## [106,]    5    7    9
## [107,]    5    7   10
## [108,]    5    8    9
## [109,]    5    8   10
## [110,]    5    9   10
## [111,]    6    7    8
## [112,]    6    7    9
## [113,]    6    7   10
## [114,]    6    8    9
## [115,]    6    8   10
## [116,]    6    9   10
## [117,]    7    8    9
## [118,]    7    8   10
## [119,]    7    9   10
## [120,]    8    9   10
```

```
total_combinaciones <- nrow(grupos_posibles)
cat("Cantidad total de combinaciones posibles:", total_combinaciones,
"\n")
```

```
## Cantidad total de combinaciones posibles: 120
```

#Probabilidad Condicional

```r
total_figuras <- 12
cantidad_reyes <- 4

prob_rey_dado_figura <- cantidad_reyes / total_figuras
cat("Probabilidad de que Carlos saque un rey si sabe que es una figura:",
prob_rey_dado_figura, "\n")
```

```
## Probabilidad de que Carlos saque un rey si sabe que es una figura:
0.3333333
```

#Teorema de Bayes

```r
P_magico <- 0.3
P_normal <- 0.7

P_brilla_magico <- 0.8
P_brilla_normal <- 0.1

P_brilla <- (P_magico * P_brilla_magico) + (P_normal * P_brilla_normal)
P_magico_dado_brilla <- (P_magico * P_brilla_magico) / P_brilla

cat("Probabilidad de que un árbol que brilla sea mágico:",
round(P_magico_dado_brilla * 100, 2), "%\n")
```

```
## Probabilidad de que un árbol que brilla sea mágico: 77.42 %
```

#Distribución Binomial 1

```r
n_habitantes <- 5
prob_tea <- 0.6
prob_exacto_3 <- dbinom(3, size = n_habitantes, prob = prob_tea)

cat("Probabilidad de que exactamente 3 personas prefieran el te:",
prob_exacto_3, "\n")
```

```
## Probabilidad de que exactamente 3 personas prefieran el te: 0.3456
```

Distribución Binomial 2

```r
n_total <- 10
prob_tea_2 <- 0.7
prob_exacto_7 <- dbinom(7, size = n_total, prob = prob_tea_2)

cat("Probabilidad de que exactamente 7 personas prefieran el te:",
prob_exacto_7, "\n")
```

```
## Probabilidad de que exactamente 7 personas prefieran el te: 0.2668279
```

##Distribucion de Poisson

##En el Bosque de la Inferencia, los árboles mágicos brillan en promedio 3 veces por noche. ¿Cuál es la probabilidad de que un árbol brille exactamente 5 veces en una noche?

```
# Tasa promedio de eventos (brillos por noche)
lambda <- 3
# Eventos deseados
k <- 5

# Cálculo
prob_poisson <- dpois(k, lambda)

# Resultado
cat("La probabilidad de que un arbol brille 5 veces es:",
round(prob_poisson, 3), "\n")
```

```
## La probabilidad de que un arbol brille 5 veces es: 0.101
```

##Distribucion Normal

##Las alturas de los habitantes del Pueblo de los Datos siguen una distribución normal con media μ = 170 cm y desviación estándar σ = 10 cm. ¿Cuál es la probabilidad de que un habitante mida menos de 160 cm?

```
media <- 170
#Desviacion estandar
desviacion <- 10

#Altura menor a
altura_limite <- 160

# Cálculo
prob_normal <- pnorm(altura_limite, mean = media, sd = desviacion)

# Resultado
cat("La probabilidad de que un habitante tenga una altura menor a 160 cm
es:", round(prob_normal, 3), "\n")
```

```
## La probabilidad de que un habitante tenga una altura menor a 160 cm
es: 0.159
```

##Distribucion Exponencial

##El tiempo entre llegadas de visitantes al Bosque de la Inferencia sigue una distribución exponencial con una tasa de λ = 0.5 visitantes por minuto. ¿Cuál es la probabilidad de que el próximo visitante llegue en menos de 2 minutos?

```
# Tasa de λ
tasa <- 0.5
```

```r
#Tiempo limite
tiempo_limite <- 2

# Cálculo
prob_exponencial <- pexp(tiempo_limite, rate = tasa)

# Resultado
cat("La probabilidad de que el proximo visitante llegue en menos de 2
minutos es:", round(prob_exponencial, 3), "\n")
```

```
## La probabilidad de que el proximo visitante llegue en menos de 2
minutos es: 0.632
```

##Distribución Uniforme

##En el Pueblo de los Datos, el tiempo que tarda un mensajero en entregar una carta sigue una distribución uniforme entre 10 y 30 minutos. ¿Cuál es la probabilidad de que el mensajero tarde menos de 20 minutos?

```r
# Minimo
minimo <- 10

#Maximo
maximo <- 30

#Limite de tiempo
limite_superior <- 20

# Cálculo
prob_uniforme <- punif(limite_superior, min = minimo, max = maximo)

# Resultado
cat("La probabilidad de que el mensajero haga la entrega en menos de  20
minutos es:", round(prob_uniforme, 3), "\n")
```

```
## La probabilidad de que el mensajero haga la entrega en menos de  20
minutos es: 0.5
```

#Sección 2: Muestreo ###Muestreo aleatorio simple

```r
set.seed(123)
poblacion <- 1:1000
muestra <- sample(poblacion, 100)

print(muestra)
```

```
##   [1] 415 463 179 526 195 938 818 118 299 229 244  14 374 665 602 603
768 709
##  [19]  91 953 348 649 355 840  26 519 426 979 766 211 932 590 593 555
871 373
```

```
##  [37] 844 143 544 490 621 775 905 937 842  23 923 309 135 821 954 224
166 217
##  [55] 290 581  72 588 575 141 722 865 859 153 294 277 999  41 431  90
316 223
##  [73] 528 116 606 774 747 456 598 854  39 159 752 209 988 994  34 516
13  69
##  [91] 895 755 409 308 278  89 537 291 424 880
```

###Explicacion

#1(POBLACION)Estos serian los 1000 habitantes #2(MUESTRA)Seria la seleccion aleatoria de 100 personas

###Muestreo Estratificado # Muestreo estratificado con dplyr

```r
# Establecer semilla para reproducibilidad
set.seed(123)

# Definir cantidad por barrio
n_por_barrio <- 250

# Crear vector con nombres de barrios
barrios <- rep(c("Norte", "Sur", "Este", "Oeste"), each = n_por_barrio)

# Crear la población como data frame
poblacion <- data.frame(ID = 1:1000, Barrio = barrios)

# Cargar librería necesaria
library(dplyr)

# Realizar muestreo estratificado: 25 observaciones por barrio
muestra_estratificada <- poblacion %>%
  group_by(Barrio) %>%
  slice_sample(n = 25)

# Mostrar la muestra
print(muestra_estratificada)
```

```
## # A tibble: 100 × 2
## # Groups:   Barrio [4]
##       ID Barrio
##    <int> <chr>
##  1   659 Este
##  2   707 Este
##  3   679 Este
##  4   514 Este
##  5   695 Este
##  6   670 Este
##  7   550 Este
##  8   618 Este
```

```
##  9   543 Este
## 10   729 Este
## # ℹ 90 more rows
```

###Explicación: #1Se crea una población con su barrio correspondiente. #2(group_by(Barrio)) agrupa por barrio #3(slice_sample(n = 25)) tomara 25 muestras aleatorias por grupo.

###Tamano de la muestra

```
#Fórmula para tamano de muestra para proporciones

# Parámetros para calcular tamaño de muestra
nivel_confianza <- 0.95
margen_error <- 0.05
p <- 0.5  # Proporción esperada (máxima varianza)

# Valor Z correspondiente al nivel de confianza
z <- qnorm(1 - (1 - nivel_confianza) / 2)

# Cálculo del tamaño de muestra necesario para una proporcion
n <- (z^2 * p * (1 - p)) / (margen_error^2)

# Mostrar resultado redondeado hacia arriba
cat("Tamano de muestra necesario:", ceiling(n), "\n")
```

```
## Tamano de muestra necesario: 385
```

###Explicacion #P Sera el valor esperado #Se usa la fórmula estándarpara estimar proporciones. #1.(qnorm()) da el valor Z para el nivel devconfianza. #2.(p = 0.5) maximiza el tamaño necesario (caso mas conservador). #3.(ceiling()) redondea al entero superior.

###Muestreo de conglomerados # Muestreo por conglomerados

```
set.seed(123)

# Crear población con 10 distritos (cada uno con 100 personas)
distritos <- rep(1:10, each = 100)
poblacion <- data.frame(ID = 1:1000, Distrito = distritos)

# Seleccionar 2 distritos al azar
distritos_seleccionados <- sample(1:10, 2)

# Filtrar la muestra: solo individuos de los distritos seleccionados
muestra_conglomerados <- subset(poblacion, Distrito %in%
distritos_seleccionados)

# Mostrar la muestra obtenida
print(muestra_conglomerados)
```

```
##        ID Distrito
## 201   201        3
## 202   202        3
## 203   203        3
## 204   204        3
## 205   205        3
## 206   206        3
## 207   207        3
## 208   208        3
## 209   209        3
## 210   210        3
## 211   211        3
## 212   212        3
## 213   213        3
## 214   214        3
## 215   215        3
## 216   216        3
## 217   217        3
## 218   218        3
## 219   219        3
## 220   220        3
## 221   221        3
## 222   222        3
## 223   223        3
## 224   224        3
## 225   225        3
## 226   226        3
## 227   227        3
## 228   228        3
## 229   229        3
## 230   230        3
## 231   231        3
## 232   232        3
## 233   233        3
## 234   234        3
## 235   235        3
## 236   236        3
## 237   237        3
## 238   238        3
## 239   239        3
## 240   240        3
## 241   241        3
## 242   242        3
## 243   243        3
## 244   244        3
## 245   245        3
## 246   246        3
## 247   247        3
## 248   248        3
## 249   249        3
```

```
## 250   250         3
## 251   251         3
## 252   252         3
## 253   253         3
## 254   254         3
## 255   255         3
## 256   256         3
## 257   257         3
## 258   258         3
## 259   259         3
## 260   260         3
## 261   261         3
## 262   262         3
## 263   263         3
## 264   264         3
## 265   265         3
## 266   266         3
## 267   267         3
## 268   268         3
## 269   269         3
## 270   270         3
## 271   271         3
## 272   272         3
## 273   273         3
## 274   274         3
## 275   275         3
## 276   276         3
## 277   277         3
## 278   278         3
## 279   279         3
## 280   280         3
## 281   281         3
## 282   282         3
## 283   283         3
## 284   284         3
## 285   285         3
## 286   286         3
## 287   287         3
## 288   288         3
## 289   289         3
## 290   290         3
## 291   291         3
## 292   292         3
## 293   293         3
## 294   294         3
## 295   295         3
## 296   296         3
## 297   297         3
## 298   298         3
## 299   299         3
```

```
## 300   300        3
## 901   901       10
## 902   902       10
## 903   903       10
## 904   904       10
## 905   905       10
## 906   906       10
## 907   907       10
## 908   908       10
## 909   909       10
## 910   910       10
## 911   911       10
## 912   912       10
## 913   913       10
## 914   914       10
## 915   915       10
## 916   916       10
## 917   917       10
## 918   918       10
## 919   919       10
## 920   920       10
## 921   921       10
## 922   922       10
## 923   923       10
## 924   924       10
## 925   925       10
## 926   926       10
## 927   927       10
## 928   928       10
## 929   929       10
## 930   930       10
## 931   931       10
## 932   932       10
## 933   933       10
## 934   934       10
## 935   935       10
## 936   936       10
## 937   937       10
## 938   938       10
## 939   939       10
## 940   940       10
## 941   941       10
## 942   942       10
## 943   943       10
## 944   944       10
## 945   945       10
## 946   946       10
## 947   947       10
## 948   948       10
## 949   949       10
```

```
## 950   950       10
## 951   951       10
## 952   952       10
## 953   953       10
## 954   954       10
## 955   955       10
## 956   956       10
## 957   957       10
## 958   958       10
## 959   959       10
## 960   960       10
## 961   961       10
## 962   962       10
## 963   963       10
## 964   964       10
## 965   965       10
## 966   966       10
## 967   967       10
## 968   968       10
## 969   969       10
## 970   970       10
## 971   971       10
## 972   972       10
## 973   973       10
## 974   974       10
## 975   975       10
## 976   976       10
## 977   977       10
## 978   978       10
## 979   979       10
## 980   980       10
## 981   981       10
## 982   982       10
## 983   983       10
## 984   984       10
## 985   985       10
## 986   986       10
## 987   987       10
## 988   988       10
## 989   989       10
## 990   990       10
## 991   991       10
## 992   992       10
## 993   993       10
## 994   994       10
## 995   995       10
## 996   996       10
## 997   997       10
## 998   998       10
```

```
## 999   999       10
## 1000 1000       10
```

###Explicacion #1.Creamos la poblacion #2.Seleccionamos los distritosaleatoriamente #3>Tenemos que filtrar la muestra con la formulaasignada

###Muestreo sistematico

```
# Muestreo sistemático

set.seed(123)

# Crear población del 1 al 1000
poblacion <- 1:1000

# Establecer el intervalo de selección
intervalo <- 20

# Seleccionar aleatoriamente el punto de inicio entre 1 y 20
inicio <- sample(1:intervalo, 1)

# Generar las posiciones sistemáticas
posiciones <- seq(inicio, length.out = 50, by = intervalo)

# Obtener la muestra
muestra_sistematica <- poblacion[posiciones]

# Mostrar resultados
cat("Punto de inicio:", inicio, "\n")

## Punto de inicio: 15

print(muestra_sistematica)

##  [1]  15  35  55  75  95 115 135 155 175 195 215 235 255 275 295 315
335 355 375
## [20] 395 415 435 455 475 495 515 535 555 575 595 615 635 655 675 695
715 735 755
## [39] 775 795 815 835 855 875 895 915 935 955 975 995
```

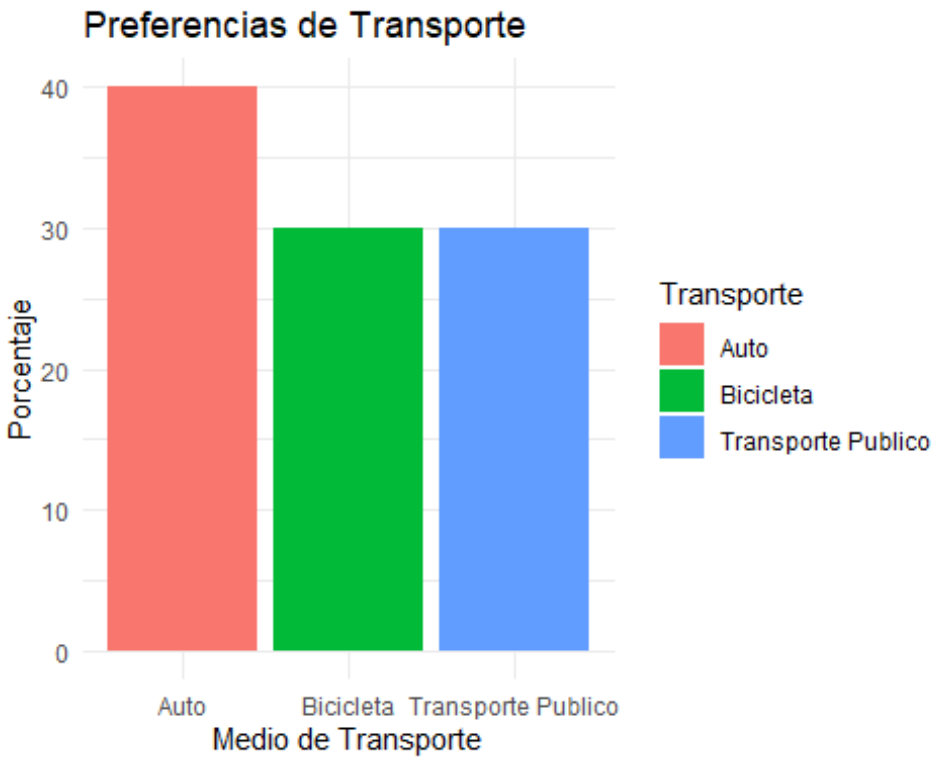
Sección 3: Storytelling con Datos

##Sección 3: Storytelling con Datos

#Gráfico de Barras: Preferencias de Transporte

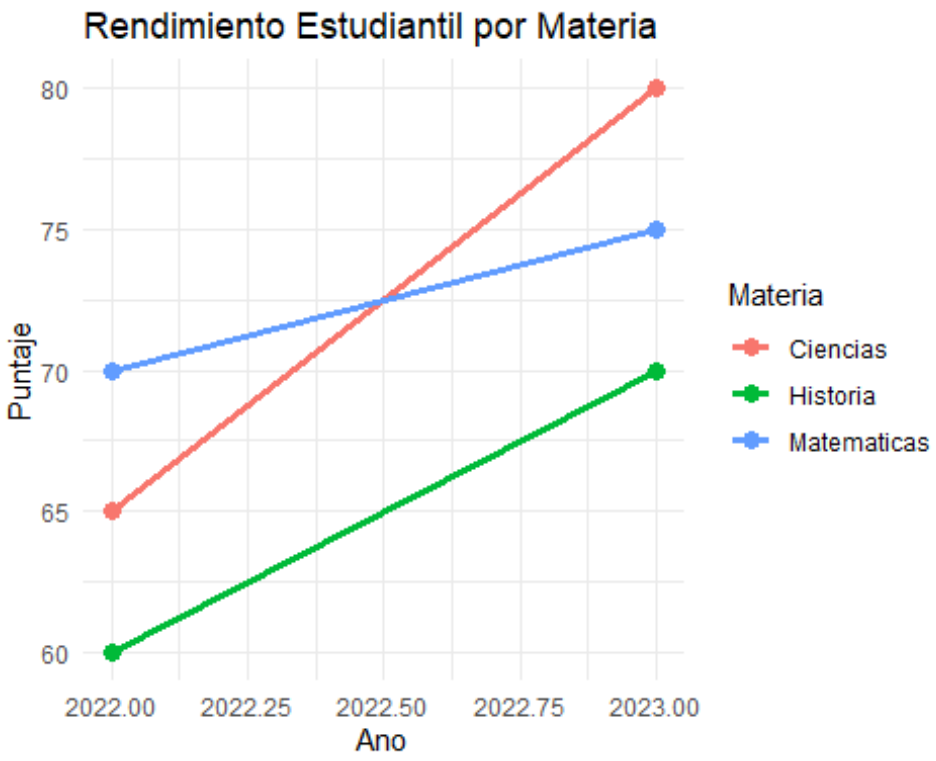
```
preferencias <- data.frame(
  Transporte = c("Auto", "Bicicleta", "Transporte Publico"),
  Porcentaje = c(40, 30, 30)
)
```

```r
ggplot(preferencias, aes(x = Transporte, y = Porcentaje, fill =
Transporte)) +
  geom_bar(stat = "identity") +
  labs(title = "Preferencias de Transporte", y = "Porcentaje", x = "Medio
de Transporte") +
  theme_minimal()
```



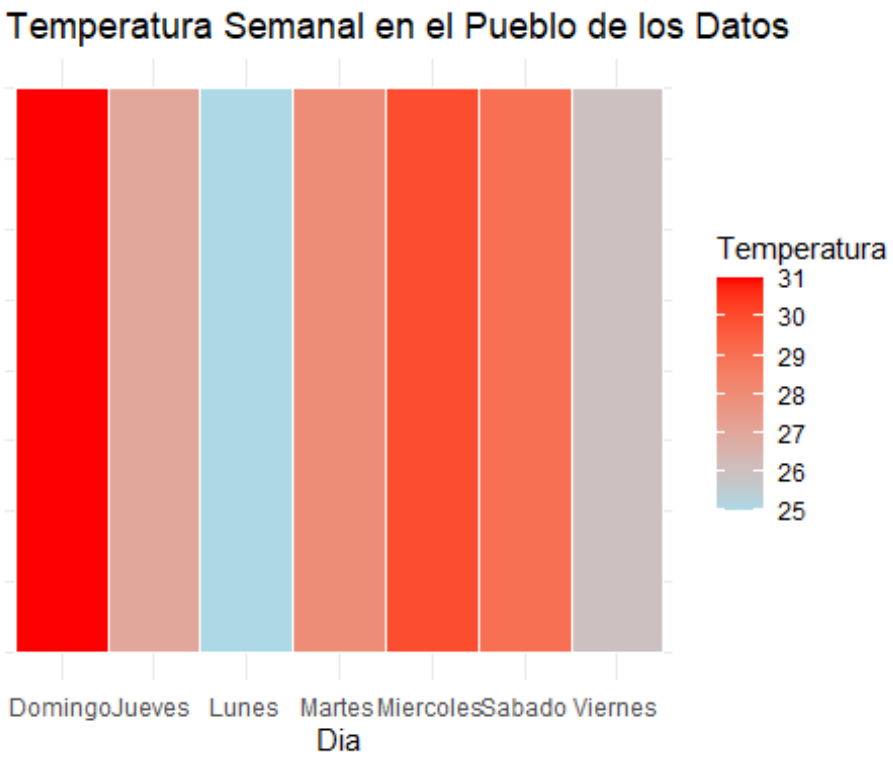
##Gráfico de Líneas: Rendimiento Estudiantil

```r
datos_rendimiento <- data.frame(
  Ano = rep(c(2022, 2023), 3),
  Materia = rep(c("Matematicas", "Ciencias", "Historia"), each = 2),
  Rendimiento = c(70, 75, 65, 80, 60, 70)
)

ggplot(datos_rendimiento, aes(x = Ano, y = Rendimiento, group = Materia,
color = Materia)) +
  geom_line(size = 1.2) +
  geom_point(size = 3) +
  labs(title = "Rendimiento Estudiantil por Materia", y = "Puntaje", x =
"Ano") +
  theme_minimal()
```



##Mapa de Calor: Temperaturas Semanales

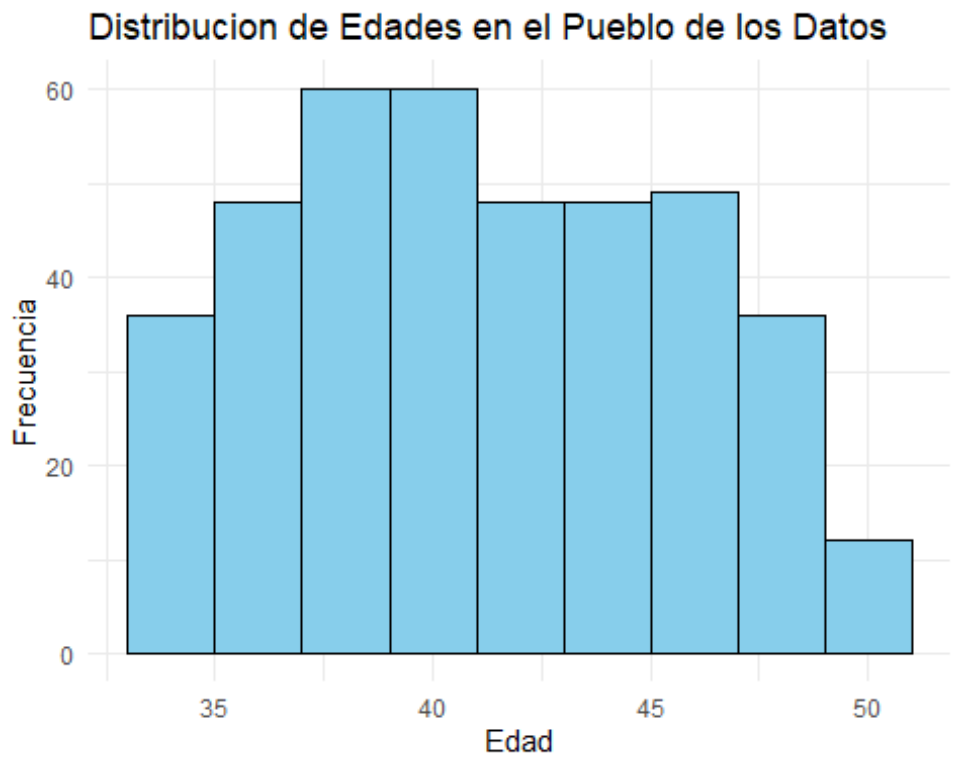
```
datos_temp <- data.frame(
  Dia = c("Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado",
"Domingo"),
  Temperatura = c(25, 28, 30, 27, 26, 29, 31)
)

ggplot(datos_temp, aes(x = Dia, y = 1, fill = Temperatura)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightblue", high = "red") +
  labs(title = "Temperatura Semanal en el Pueblo de los Datos", y = "", x
= "Dia") +
  theme_minimal() +
  theme(axis.text.y = element_blank(), axis.ticks.y = element_blank())
```



##Histograma: Distribución de Edades

```
edades <- c(42, 39, 46, 37, 48, 34, 50, 38, 45, 40, 43, 36, 47, 35, 49,
41, 38, 44, 40, 46,
            37, 42, 39, 48, 36, 45, 41, 43, 35, 47, 38, 44, 40, 46, 37,
42, 39, 48, 34, 50,
            38, 45, 40, 43, 36, 47, 35, 49, 41, 38, 44, 40, 46, 37, 42,
39, 48, 36, 45, 41,
            43, 35, 47, 38, 44, 40, 46, 37, 42, 39, 48, 34, 50, 38, 45,
40, 43, 36, 47, 35,
            49, 41, 38, 44, 40, 46, 37, 42, 39, 48, 36, 45, 41, 43, 35,
47, 38, 44, 40, 46,
            37, 42, 39, 48, 34, 50, 38, 45, 40, 43, 36, 47, 35, 49, 41,
38, 44, 40, 46, 37,
            42, 39, 48, 36, 45, 41, 43, 35, 47, 38, 44, 40, 46, 37, 42,
39, 48, 34, 50, 38,
            45, 40, 43, 36, 47, 35, 49, 41, 38, 44, 40, 46, 37, 42, 39,
48, 36, 45, 41, 43,
            35, 47, 38, 44, 40, 46, 37, 42, 39, 48, 34, 50, 38, 45, 40,
43, 36, 47, 35, 49,
            41, 38, 44, 40, 46, 37, 42, 39, 48, 36, 45, 41, 43, 35, 47,
38, 44, 40, 46, 37,
            42, 39, 48, 34, 50, 38, 45, 40, 43, 36, 47, 35, 49, 41, 38,
44, 40, 46, 37, 42,
            39, 48, 36, 45, 41, 43, 35, 47, 38, 44, 40, 46, 37, 42, 39,
48, 34, 50, 38, 45,
            40, 43, 36, 47, 35, 49, 41, 38, 44, 40, 46, 37, 42, 39, 48,
36, 45, 41, 43, 35,
```

```
            47, 38, 44, 40, 46, 37, 42, 39, 48, 34, 50, 38, 45, 40, 43,
36, 47, 35, 49, 41,
            38, 44, 40, 46, 37, 42, 39, 48, 36, 45, 41, 43, 35, 47, 38,
44, 40, 46, 37, 42,
            39, 48, 34, 50, 38, 45, 40, 43, 36, 47, 35, 49, 41, 38, 44,
40, 46, 37, 42, 39,
            48, 36, 45, 41, 43, 35, 47, 38, 44, 40, 46, 37, 42, 39, 48,
34, 50, 38, 45, 40,
            43, 36, 47, 35, 49, 41, 38, 44, 40, 46, 37, 42, 39, 48, 36,
45, 41, 43, 35, 47,
            38, 44, 40, 46, 37, 42, 39, 48, 34, 50, 38, 45, 40, 43, 36,
47, 35, 49, 41, 38,
            44, 40, 46, 37, 42, 39, 48, 36, 45, 41, 43, 35, 47, 38, 44,
40, 46)

ggplot(data.frame(edades), aes(x = edades)) +
  geom_histogram(binwidth = 2, fill = "skyblue", color = "black") +
  labs(title = "Distribucion de Edades en el Pueblo de los Datos", x =
"Edad", y = "Frecuencia") +
  theme_minimal()
```




```
```