Medio CDMX

Sonia Mancera

2024-07-20

```
file_list <- list.files(path = "./data", pattern = "*.csv", full.names = TRUE)

data <- file_list %>%
  set_names() %>%
  map_dfr(~ read_csv(.x) %>% mutate(filename = basename(.x)), .id = "categoria") %>% clean_names() %>%
  mutate(categoria = str_remove_all(categoria, './data/'))
```

```
## Rows: 6282 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Nombre, Paso por Km, Velocidad Media
## dbl  (2): ...1, BIB
## time (6): Tiempo Oficial, Tiempo Chip, Split 5k, Split 10k, Split 15k, Split...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 6212 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Nombre, Paso por Km, Velocidad Media
## dbl  (2): ...1, BIB
## time (6): Tiempo Oficial, Tiempo Chip, Split 5k, Split 10k, Split 15k, Split...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## New names:
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 2490 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Nombre, Paso por Km, Velocidad Media
## dbl  (2): ...1, BIB
## time (6): Tiempo Oficial, Tiempo Chip, Split 5k, Split 10k, Split 15k, Split...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Data wrangling (not the best practice to put it here but fully transparent with the code)

algunos tiempos no son validos, los limpiamos

```
# data <- read_csv('data/libre_varonil.csv')
#
#
# is_valid_time <- function(x) {
#   !is.na(parse_date_time(x, orders = c("HMS", "HM", "MS", "H", "M", "S")))
# }
#
# # Filter out rows with invalid 'Split 5k' entries and convert valid ones
# data %>%
#   filter(is_valid_time(`Split 5k`)) %>%
#   mutate(`Split 5k` = hms::as_hms(`Split 5k`)) %>%
#   write_csv('data/libre_varonil.csv')
```

Procesamiento de datos

```
corrales <- tribble(~BLOQUE,~MARGEN,
                    'ELITE',    0 ,
                    'ADIZERO',  0 ,
                    'AMARILLO', 5*60,
                    'VERDE',    15*60,
                    'AZUL',     40*60,
                    'NARANJA',  60*60)
```

```
data <- data %>% mutate(dif = tiempo_oficial - tiempo_chip,
       corral_verdadero = case_when(
        tiempo_chip < hms("01:20:00") ~ 'AMARILLO',
        tiempo_chip >= hms("01:20:00") & tiempo_chip < hms("01:40:00") ~ 'VERDE',
        tiempo_chip >= hms("01:40:00") & tiempo_chip < hms("02:00:00") ~ 'AZUL',
        tiempo_chip >= hms("02:00:00") ~ 'NARANJA'
      ),
      corral_salida = case_when(
        dif < 300 ~ 'NEGRO',
        dif >= 300 & dif < 900 ~ 'AMARILLO',
        dif >= 900 & dif < 2400 ~ 'VERDE',
        dif >= 2400 & dif < 3600 ~ 'AZUL',
        TRUE ~ 'NARANJA'
      ))
```

quitamos NA's por los chips que pudieron haber fallado o la gente que no terminó

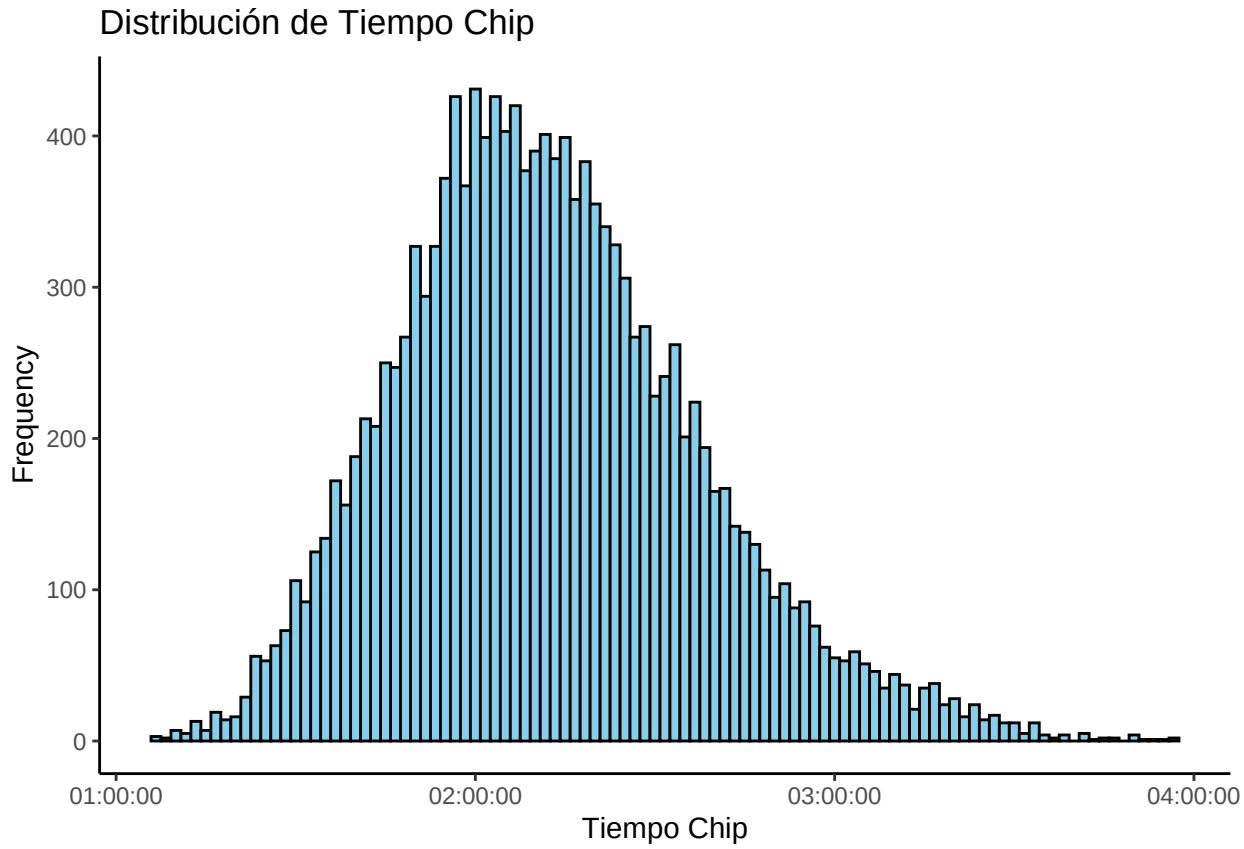
```
data <- data %>% drop_na()
```

Análisis:

Nota: hay muchos errores de chip y la neta no lo quiero corregir, va más allá de este estudio.

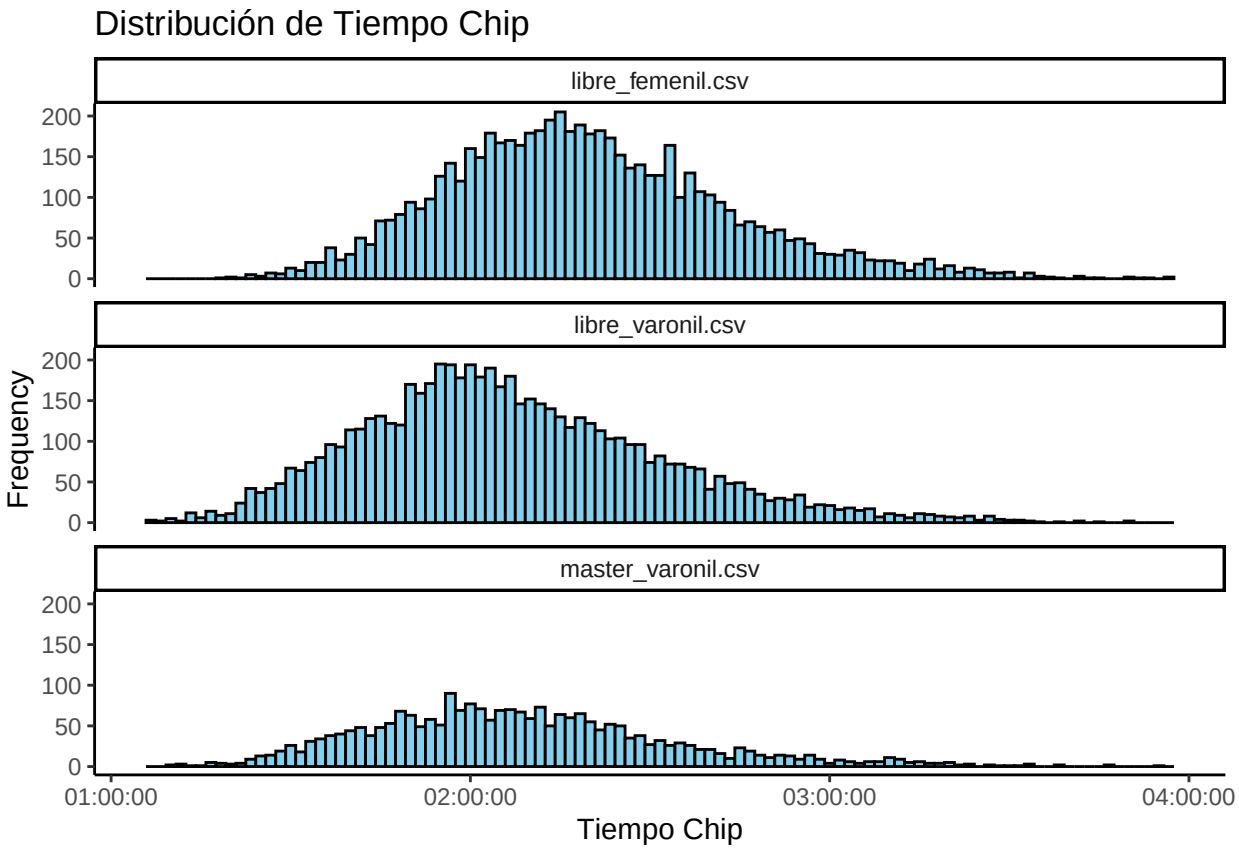
Distribución de data

```
ggplot(data, aes(x = tiempo_chip)) +
  geom_histogram(binwidth = 100, fill = "skyblue", color = "black") +
  labs(title = "Distribución de Tiempo Chip",
       x = "Tiempo Chip",
       y = "Frequency") +
  theme_classic()
```



Por categoria

```
ggplot(data, aes(x = tiempo_chip)) +
  geom_histogram(binwidth = 100, fill = "skyblue", color = "black") +
  labs(title = "Distribución de Tiempo Chip",
       x = "Tiempo Chip",
       y = "Frequency") +
  theme_classic() +
  facet_wrap(~categoria, ncol = 1)
```



Supuestos: voy a ser una ciudadana que confía en la buena fé de la gente y asumiremos que los que salieron de Elite + Adizero si van ahí

```
data <- data %>% mutate(corral_verdadero = ifelse(corral_salida == 'NEGRO', 'NEGRO', corral_verdadero))
```

Plot

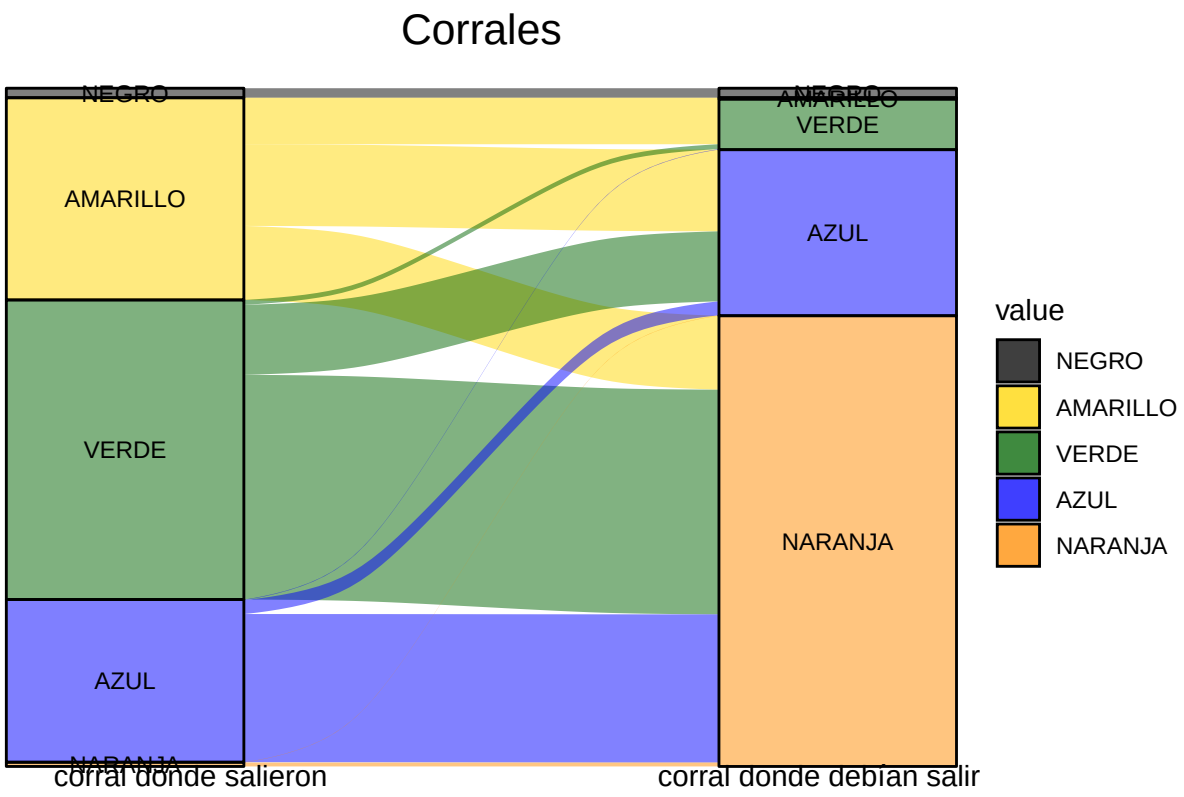
```
levels_order <- c("NEGRO", "AMARILLO", "VERDE", "AZUL", "NARANJA")

data_to_plot <- data %>%
  group_by(corral_verdadero, corral_salida) %>%
  count() %>%
  rownames_to_column('index') %>%
  pivot_longer(-c(index, n))

data_to_plot <- data_to_plot %>%
  mutate(
    value = factor(value, levels = levels_order)
)

ggplot(data_to_plot,
       aes(x = name, stratum = value, alluvium = index,
           y = n,
           fill = value, label = value)) +
  scale_x_discrete(expand = c(.1, .1)) +
  scale_fill_manual(values = c("NEGRO" = "black", "AMARILLO" = "#FFD700", "VERDE" = "darkgreen", "AZUL"
```

```
  geom_flow(stat = "alluvium", lode.guidance = "frontback") +
  geom_stratum(alpha = .5) +
  geom_text(stat = "stratum", aes(label = after_stat(stratum)), size = 3) +
  theme_void() +
  theme(
    axis.title = element_blank(),
    axis.text = element_blank(),
    axis.ticks = element_blank(),
    panel.grid = element_blank(),
    plot.title = element_text(hjust = 0.5, size = 16),
    plot.background = element_rect(fill = "white", color = NA),
    plot.margin = margin(10, 10, 10, 10)
  ) +
  ggtitle("Corrales") +
  annotate("text", x = .9, y = -200, label = "corral donde salieron", size = 4, hjust = 0) +
  annotate("text", x = 2.2, y = -200, label = "corral donde debían salir", size = 4, hjust = 1)
```




```
data %>%
  group_by(corral_salida) %>%
  count() %>%
  ungroup() %>%
  mutate(total = sum(n),
         porc = n/total)
```

```
## # A tibble: 5 x 4
```

```
##   corral_salida     n total    porc
##   <chr>         <int> <int>   <dbl>
## 1 AMARILLO       4370 14662 0.298
## 2 AZUL           3517 14662 0.240
## 3 NARANJA          92 14662 0.00627
## 4 NEGRO           206 14662 0.0140
## 5 VERDE          6477 14662 0.442
```

```
data %>%
  group_by(corral_verdadero) %>%
  count() %>%
  ungroup() %>%
  mutate(total = sum(n),
         porc = n/total)
```

```
## # A tibble: 5 x 4
##   corral_verdadero     n total    porc
##   <chr>            <int> <int>   <dbl>
## 1 AMARILLO            28 14662 0.00191
## 2 AZUL              3590 14662 0.245
## 3 NARANJA           9744 14662 0.665
## 4 NEGRO              206 14662 0.0140
## 5 VERDE             1094 14662 0.0746
```