FRP y FRM

1) Introducción

En esta práctica se construirá la función de regresión poblacional, posteriormente se sacarán múltiples muestras con las cuales se analizará la variabilidad de la muestra por medio de la función de regresión muestral. Es importante recalcar que comúnmente los datos poblacionaes no son conocidos, pero en este ejemplo supondremos que los datos que nos proporcionan son los poblacionales.

Se importan los datos de "marketing" que contienen datos de inversión en tres tipos de medios publicitarios y datos de las ventas:

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(datarium)
library(tidymodels)  # package that imports useful packages for modeling
library(readr)       # for importing data
library(dotwhisker)  # visualize regression results

data("marketing", package = "datarium")

marketing <-
  marketing %>%
  gather(company, investment, -sales)

marketing %>%
  head(n=6)
```

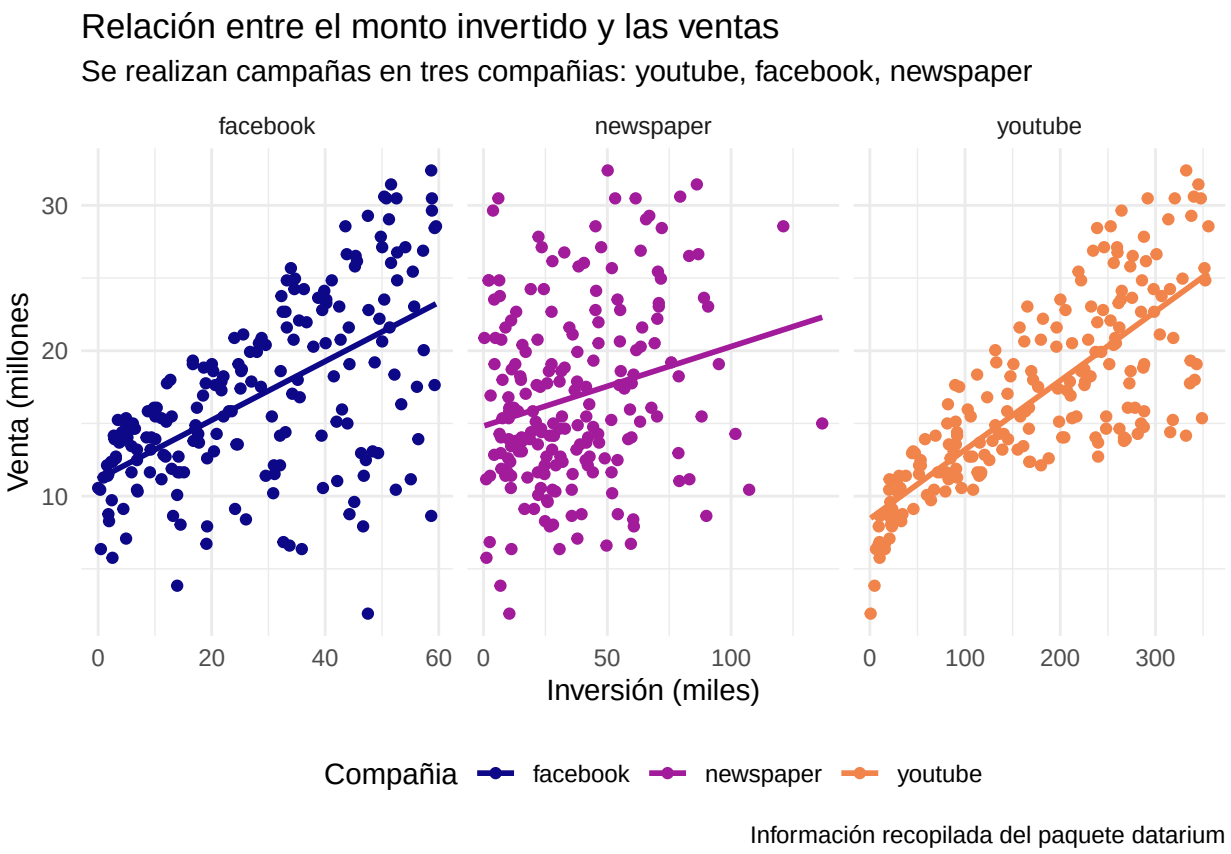
```
##   sales company investment
## 1 26.52 youtube     276.12
## 2 12.48 youtube      53.40
## 3 11.16 youtube      20.64
## 4 22.20 youtube     181.80
## 5 15.48 youtube     216.96
## 6  8.64 youtube      10.44
```

1.1 Exploratory Data Analysis

Nos gustaría conocer como se comportan las ventas con respecto a la inversión en distintos medios. Para esto, es conveniente graficar en facets para cada medio:

```
marketing %>%
  ggplot(aes(x = investment, y = sales, group = company, col = company)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE) +
  facet_wrap(~company, scales = "free_x")+
  theme_minimal()+
```

```
  theme(legend.position = "bottom")+
  labs(title = "Relación entre el monto invertido y las ventas",
       subtitle = "Se realizan campañas en tres compañias: youtube, facebook, newspaper",
       caption = "Información recopilada del paquete datarium",
       color = "Compañia",
       x = "Inversión (miles)",
       y = "Venta (millones")+
  scale_color_viridis_d(option = "plasma", end = .7)
```



1.2 Regresión lineal para compañia de Youtube

Por fines ilustrativos, se realizará la regresión lineal solo para datos de youtube. Para esto se ocupa la funciones de linear_reg() y fit():

```
marketing_youtube <- marketing %>% filter(company == "youtube")
# 1 - entrenar modelo
lm_fit <-
  linear_reg() %>%
  fit(sales ~ investment, data = marketing_youtube)

# 2 - visualizar y graficar
glance(lm_fit)
```

```
## # A tibble: 1 x 12
```

```
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.612         0.610  3.91      312. 1.47e-42     1  -556. 1117. 1127.
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
tidy(lm_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   8.44     0.549        15.4 1.41e-35
## 2 investment    0.0475   0.00269      17.7 1.47e-42
```

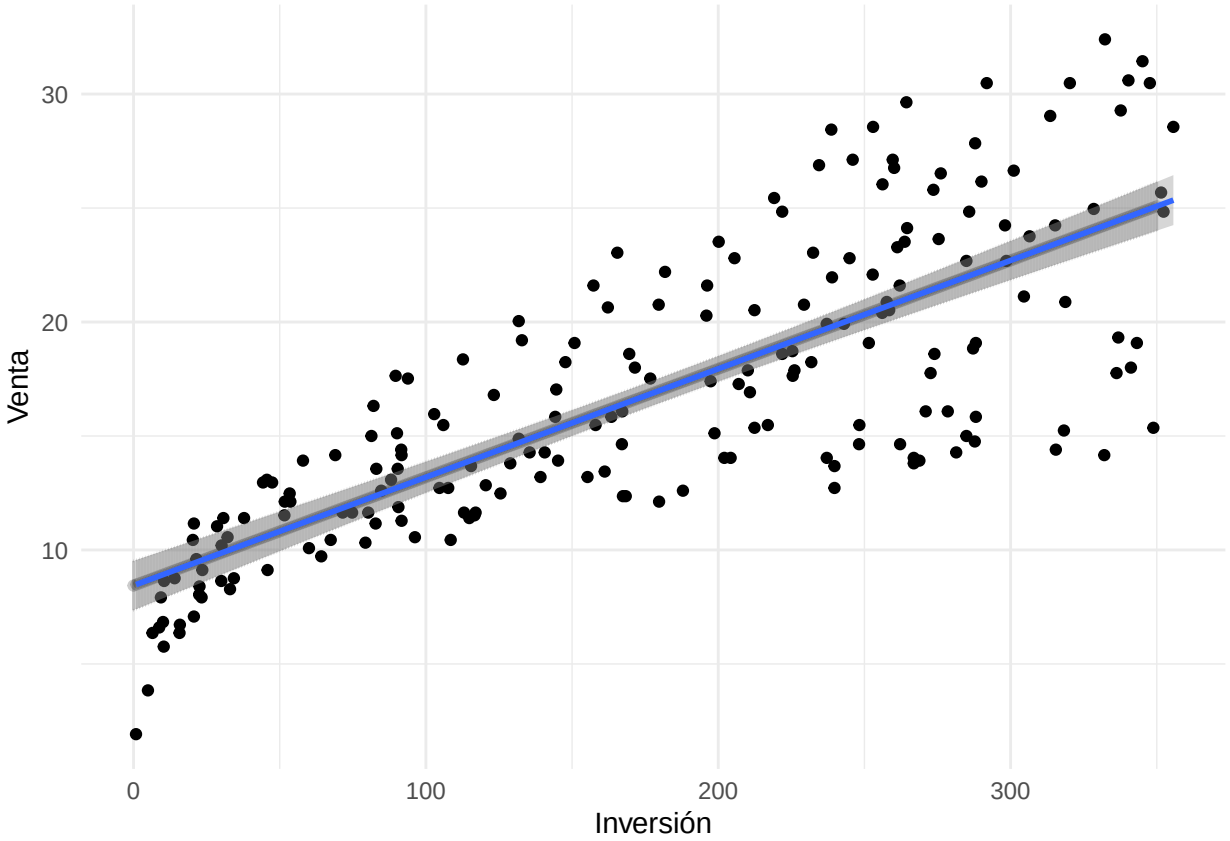
1.3 Funcion de Regresión Poblacional (FRP)

¿Cómo se ve la recta de regresión lineal para distintos niveles de X? Para esto, se realiza un grid de 0 a 350 y se realiza la predicción usando el modelo lineal. Esto nos calculará para cada punto x_i el valor de $\hat{y} = \beta_0 + \beta_1 x_i$

```
new_points <- expand.grid(investment = 0:350)
mean_pred <- predict(lm_fit, new_data = new_points)
conf_int_pred <-
  predict(
    object = lm_fit,
    new_data = new_points,
    type = "conf_int",
    level = .95
  )

plot_data <-
  new_points %>%
  bind_cols(mean_pred) %>%
  bind_cols(conf_int_pred)

ggplot(plot_data, aes(x = investment)) +
  geom_point(data = marketing_youtube, aes(x = investment, y = sales))+
  geom_point(aes(y = .pred), alpha = 0.2) +
  geom_errorbar(aes(ymin = .pred_lower,
                    ymax = .pred_upper),
                width = .2, alpha = 0.2) +
  geom_smooth(data = marketing_youtube,aes(x = investment, y = sales), method = "lm")+
  labs(y = "Venta", x = "Inversión") +
  theme_minimal()
```



Recordemos que supondremos que los datos poblacionales son los datos poblacionales, estonces la recta obtenida es la Función de Regresión Poblacional.

1.4 Coeficientes de β_0 y β_1

Podemos calcular los coeficientes de β_0 y β_1 con las formulas:

```
# 5 - ¿cómo se ven los coeficientes con las fórmulas?
b1 <- cov(marketing_youtube$sales,marketing_youtube$investment)/var(marketing_youtube$investment)
b0 <- mean(marketing_youtube$sales)-b1*mean(marketing_youtube$investment)
b1
```

```
## [1] 0.04753664
```

```
b0
```

```
## [1] 8.439112
```

O obtenerlos directamente con la función de tidy()

```
tidy(lm_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
```

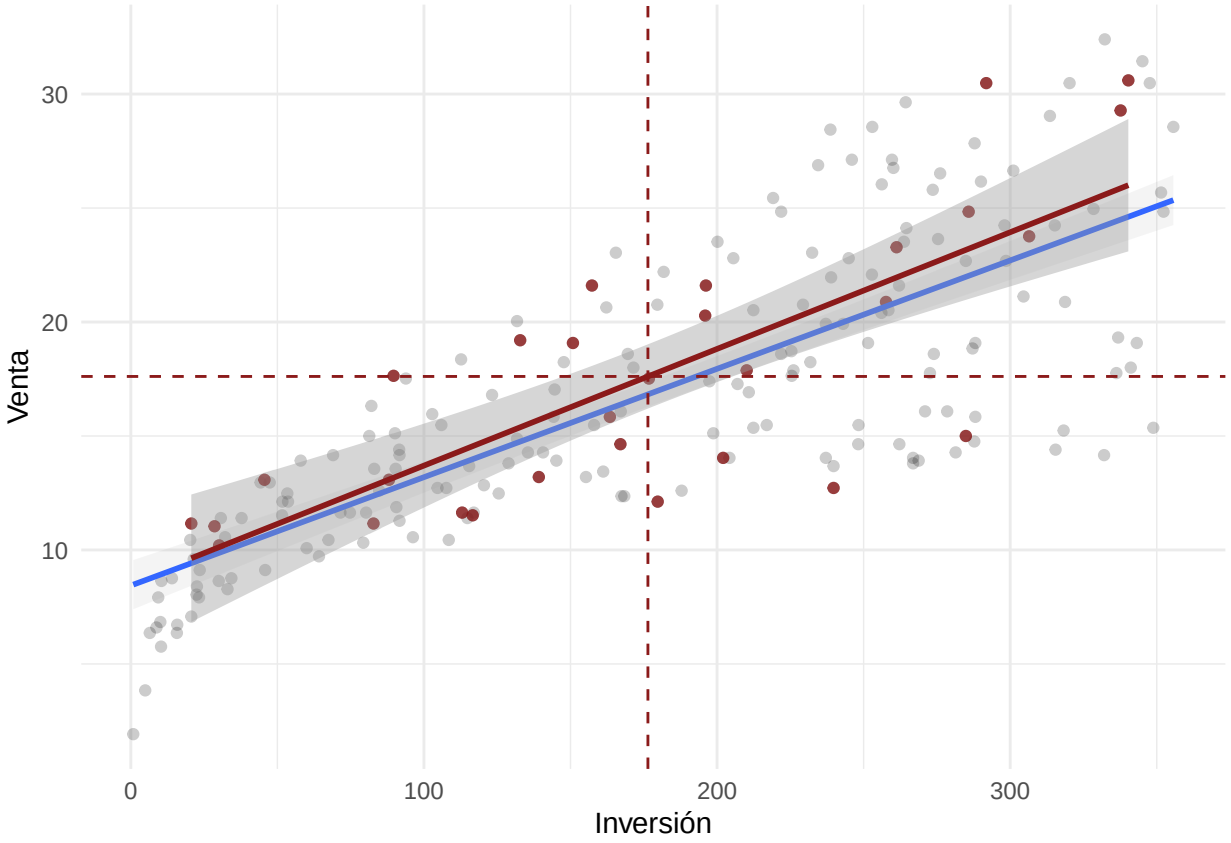
```
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   8.44     0.549        15.4 1.41e-35
## 2 investment    0.0475   0.00269      17.7 1.47e-42
```

1.5 Función de Regresión Muestral (FRM)

Ahora sacaremos una muestra de tamaño 30 y veremos la recta muestral ajustada. Si se ejecuta varias veces este código, se observará que la línea se mueve dependiendo la muestra obtenida. También, si modificamos el tamaño de muestra, se puede observar que entre más grande sea el tamaño de la muestra, menor variación de la recta de regresión muestral.

```
marketing_youtube_sample = marketing_youtube[sample(1:nrow(marketing_youtube), size = 30),]

ggplot(plot_data, aes(x = investment)) +
  geom_point(
    data = marketing_youtube,
    aes(x = investment, y = sales), alpha = 0.2)+
  geom_point(
    data = marketing_youtube_sample,
    aes(x = investment, y = sales), alpha = 0.8, color = "firebrick4")+
  geom_smooth(
    data = marketing_youtube,
    aes(x = investment, y = sales), method = "lm", alpha = 0.1)+
  geom_smooth(
    data = marketing_youtube_sample,
    aes(x = investment, y = sales), method = "lm", color = "firebrick4")+
  geom_vline(
    xintercept = mean(marketing_youtube_sample$investment),
    linetype = 2, color = "firebrick4")+
  geom_hline(
    yintercept = mean(marketing_youtube_sample$sales),
    linetype = 2, color = "firebrick4")+
  labs(y = "Venta", x = "Inversión") +
  theme_minimal()
```


30
20
10
Venta
0
100
200
300
Inversión