modelo econometrico

Ana Munoz

Pre procesamiento de datos

1. Lectura de datos y formato panel

```
df <- read.csv('df_17_21_noclean.csv') %>%
  dplyr::select(year, country, total_earnings, # orden de datos panel
         total_players,
         -iso, -code, #no aplica el modelo
         pbicap,
         gdp_gr,
         -CPI, # corrupcion
         internet,
         -elect_acc,
         desempleo,
         pea, # tech access
         -net_mig,
         life_exp, # edades
         poblacion,# people
         inflacion
        ) %>%
  arrange(country, decreasing = FALSE)

head(df, 3)
```

```
##         year country total_earnings total_players   pbicap   gdp_gr internet
## 1 2017-01-01 Albania        2868.16             2 4531.032 3.898112 62.40000
## 2 2018-01-01 Albania        1346.55             3 5287.661 4.276312 65.40000
## 3 2019-01-01 Albania       37459.64            14 5396.214 2.523541 68.55039
##   desempleo     pea life_exp poblacion inflacion
## 1     13.62 1958423   79.047   2873457  1.450732
## 2     12.30 1951044   79.184   2866376  1.472953
## 3     11.47 1937930   79.282   2854191  1.257025
```

2. Valores faltantes

- Numero de Valores faltantes por variable

```
sapply(df, function(x) sum(is.na(x)))
```

```
##           year        country total_earnings  total_players         pbicap
```

```
##              0             0             0             0            10
##         gdp_gr      internet     desempleo           pea      life_exp
##              8            54             1             0             0
##      poblacion     inflacion
##              0             8
```

- corriegiendo los NAs

```r
# Pbi faltantes
## "Cuba"  "Lebanon"  "Syrian Arab Republic"  "Venezuela"
pbicap_faltantes <- unique(df[is.na(df$pbicap), ]$country)
df <- df[!df$country %in% pbicap_faltantes, ]

# Internet: 2 faltantes -> 2018 cambodia y trinidad y tobago

### Cambodia, hueco en 2018, reemplazdo por el promedio

df[df$country=='Cambodia', 'internet'][2] <-
          (df[df$country=='Cambodia', 'internet'][1] +
             df[df$country=='Cambodia', 'internet'][3])/2

### trinidad y tobago, reemplazdo por el promedio
df[df$country=='Trinidad and Tobago', 'internet'][2] <-
          (df[df$country=='Trinidad and Tobago', 'internet'][1]+
              df[df$country=='Trinidad and Tobago', 'internet'][3])/2

# Acceso a electricidad y life expectanci solo antes del 2022
# df <- df %>%
#   filter(year < as.Date("2022-01-01"))

# EXportacion tecnologica voy a quitar a los paises que no tiene exportacion por temas politicos
##"Iran, Islamic Republic of" "United Arab Emirates" "Viet Nam"
# exp_faltantes <- unique(df[is.na(df$exp_tech), ]$country)
# df <- df[!df$country %in% exp_faltantes, ]
df <- df[!df$country %in% c("Iran", "Islamic Republic of", "United Arab Emirates", "Viet Nam"), ]

# CPI macao no tiene por temas politicos
df <- df[df$country != 'Macao', ]

# Migation Hong Kong considerado dentro del gobierno de cina
df <- df[df$country != 'Hong Kong', ]

## primero para el year que falta en korea 2020
# df[df$country=='Korea, Republic of', 'net_mig'][4] <-
#    (df[df$country=='Korea, Republic of', 'net_mig'][3]+
#       df[df$country=='Korea, Republic of', 'net_mig'][5])/2

## Mismos valores que el 2022
# df[df$country=='Mongolia', 'net_mig'][4] <- -850
# df[df$country=='Mongolia', 'net_mig'][5] <- -850

### Jugadores por poblacion por millon
df$players_ppl <- (df$total_players/df$poblacion)*1000000
```

```
# # Jugadores por poblacion por millon
# df$net_mig <- (df$net_mig/df$poblacion)*1000000


###########

# verificamos NAs, ahora no tengo NAS
sapply(df, function(x) sum(is.nan(x)))
```

```
##            year         country  total_earnings   total_players          pbicap
##               0               0               0               0               0
##          gdp_gr        internet       desempleo             pea        life_exp
##               0               0               0               0               0
##       poblacion       inflacion     players_ppl
##               0               0               0
```

3. Normalizacion con logaritmo

- valores con varianzas muy grandes
- aplico normalizacion logaritmica en algunas variables

```
df_standar <- df %>%
  mutate(across(c("total_earnings", "pbicap", "poblacion", "pea"), ~log(.)))%>%
  mutate(year = year(df$year))
```

0. Preparando los datos

- Tenemos datos panel con la siguente forma 90 paises 5 anios y estas columnas
- Nuestro panel es balanceado y corto

```
dim(table(df_standar$country,df_standar$year))
```

```
## [1] 91  6
```

```
colnames(df_standar)
```

```
##  [1] "year"           "country"        "total_earnings" "total_players"
##  [5] "pbicap"         "gdp_gr"         "internet"       "desempleo"
##  [9] "pea"            "life_exp"       "poblacion"      "inflacion"
## [13] "players_ppl"
```

- definimos las variables para el modelo

```
attach(df_standar)
Y <- cbind(total_earnings)
X <- cbind(pbicap,
           # gdp_gr,
           internet,
```

```r
          life_exp,
          # cpi,
          # elect_acc,
          pea,
          desempleo,
          poblacion,
          players_ppl,
          inflacion
          # total_players
          # net_mig
          )

df_panel <- pdata.frame(df_standar,
                        index=c('country','year'))

head(df_panel,3)
```

```
##              year country total_earnings total_players   pbicap   gdp_gr
## Albania-2017 2017 Albania       7.961426             2 8.418705 3.898112
## Albania-2018 2018 Albania       7.205301             3 8.573131 4.276312
## Albania-2019 2019 Albania      10.531019            14 8.593453 2.523541
##              internet desempleo      pea life_exp poblacion inflacion
## Albania-2017 62.40000     13.62 14.48765   79.047  14.87103  1.450732
## Albania-2018 65.40000     12.30 14.48388   79.184  14.86856  1.472953
## Albania-2019 68.55039     11.47 14.47713   79.282  14.86430  1.257025
##              players_ppl
## Albania-2017   0.6960257
## Albania-2018   1.0466178
## Albania-2019   4.9050677
```

1. Efectos Fijos

```r
fijos <- plm(Y ~ X, data=df_panel, index=c('country','year'), model= "within")
summary(fijos)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = Y ~ X, data = df_panel, model = "within", index = c("country",
##     "year"))
##
## Unbalanced Panel: n = 91, T = 5-6, N = 504
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -3.150733 -0.364174 -0.012867  0.412073  2.561268
##
## Coefficients:
##                  Estimate Std. Error t-value  Pr(>|t|)
## Xpbicap         0.3853629  0.4889433  0.7882  0.431067
## Xinternet       0.0463586  0.0084863  5.4628 8.201e-08 ***
```

```
## Xlife_exp    -0.1380331  0.0474852 -2.9069  0.003851 **
## Xpea          7.8624995  6.1237339  1.2839  0.199897
## Xdesempleo   -0.0999253  0.0380663 -2.6250  0.008992 **
## Xpoblacion    3.5450442  7.3338475  0.4834  0.629086
## Xplayers_ppl  0.0284225  0.0069179  4.1086 4.820e-05 ***
## Xinflacion    0.0148237  0.0068284  2.1709  0.030519 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    433.24
## Residual Sum of Squares: 284.05
## R-Squared:      0.34436
## Adj. R-Squared: 0.18571
## F-statistic: 26.5895 on 8 and 405 DF, p-value: < 2.22e-16
```

2. Efectos aleatorios

```r
random <- plm(Y ~ X, data=df_panel, index=c('country','year'), model= "random")
summary(random)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = Y ~ X, data = df_panel, model = "random", index = c("country",
##     "year"))
##
## Unbalanced Panel: n = 91, T = 5-6, N = 504
##
## Effects:
##                  var std.dev share
## idiosyncratic 0.7014  0.8375  0.31
## individual    1.5619  1.2498  0.69
## theta:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7129  0.7129  0.7361  0.7265  0.7361  0.7361
##
## Residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.1038 -0.3665  0.0830  0.0028  0.4978  2.2820
##
## Coefficients:
##                 Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)  -10.6294603   2.9655953 -3.5843  0.000338 ***
## Xpbicap        0.4282232   0.2054547  2.0843  0.037136 *
## Xinternet      0.0631335   0.0062340 10.1272 < 2.2e-16 ***
## Xlife_exp     -0.0695225   0.0352397 -1.9728  0.048513 *
## Xpea          -2.7532740   2.6010917 -1.0585  0.289824
## Xdesempleo    -0.0153376   0.0240543 -0.6376  0.523717
## Xpoblacion     3.8452014   2.5941073  1.4823  0.138265
## Xplayers_ppl   0.0304812   0.0059536  5.1198 3.058e-07 ***
## Xinflacion     0.0105598   0.0065714  1.6069  0.108071
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    616.83
## Residual Sum of Squares: 367.73
## R-Squared:      0.40399
## Adj. R-Squared: 0.39436
## Chisq: 360.959 on 8 DF, p-value: < 2.22e-16
```

3. MCO

```r
mco = plm(Y ~ X, data=df_panel,index=c("state", "year"), model="pooling")
summary(mco)
```

```
## Pooling Model
##
## Call:
## plm(formula = Y ~ X, data = df_panel, model = "pooling", index = c("state",
##     "year"))
##
## Unbalanced Panel: n = 91, T = 5-6, N = 504
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -6.539425 -0.639589  0.084539  0.841795  3.972309
##
## Coefficients:
##                Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)  -13.2462549   1.7039140 -7.7740 4.438e-14 ***
## Xpbicap        0.1279965   0.1463889  0.8744   0.38235
## Xinternet      0.0746053   0.0066916 11.1490 < 2.2e-16 ***
## Xlife_exp     -0.0439093   0.0263943 -1.6636   0.09683 .
## Xpea          -6.1873234   1.4675150 -4.2162 2.954e-05 ***
## Xdesempleo    -0.0012124   0.0153849 -0.0788   0.93722
## Xpoblacion     7.3444172   1.4658026  5.0105 7.568e-07 ***
## Xplayers_ppl   0.0426131   0.0048657  8.7579 < 2.2e-16 ***
## Xinflacion    -0.0120282   0.0078493 -1.5324   0.12606
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    3106.4
## Residual Sum of Squares: 1103.2
## R-Squared:      0.64486
## Adj. R-Squared: 0.63912
## F-statistic: 112.353 on 8 and 495 DF, p-value: < 2.22e-16
```

Test para escoger el mejor modelo

1. Breusch-Pagan

- H0: modelo agrupado (MCO) vs H1: efectos aleatorios

- p<0.05 entonces rechazo la Ho, por ahora el mejor modelo seria aleatorios

```
plmtest(mco, type=c("bp"))
```

```
##
##  Lagrange Multiplier Test - (Breusch-Pagan)
##
## data:  Y ~ X
## chisq = 442.16, df = 1, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

2. Hausman test

- H0: efectos aleatorios vs H1: efectos fijos
- p<0.05 entonces rechazo Ho y decido que efectos fijos es mejor

```
phtest(fijos, random)
```

```
##
##  Hausman Test
##
## data:  Y ~ X
## chisq = 47.68, df = 8, p-value = 1.137e-07
## alternative hypothesis: one model is inconsistent
```

F test

- H0: modelo agrupado (MCO) vs H1: efectos fijos
- p<0.05 entonces rechazo Ho, el mejor modelo seria efectos fijos

```
pFtest(fijos, mco)
```

```
##
##  F test for individual effects
##
## data:  Y ~ X
## F = 12.977, df1 = 90, df2 = 405, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

Regresiones

Regresieon con efectos fijos

by Country Spain

```r
df_panel$country <- relevel(df_panel$country, ref = "Spain")
regresion_country_sp = lm(Y ~ X + factor(country), data = df_panel)

# summary(regresion_country)


p_values <- summary(regresion_country_sp)$coefficients[,4]
coeficiente <- summary(regresion_country_sp)$coefficients[,1]

no_significativo <- names(p_values)[which(p_values > 0.05)]

significativo_positivos <- names(p_values)[which(p_values < 0.05 & coeficiente>0)]

significativo_negativos <- names(p_values)[which(p_values < 0.05 & coeficiente<0)]
```

Analizando significancias

```r
no_significativo
```

```
##  [1] "Xpbicap"                             "Xpea"
##  [3] "Xpoblacion"                          "factor(country)Algeria"
##  [5] "factor(country)Argentina"            "factor(country)Iraq"
##  [7] "factor(country)Korea, Republic of" "factor(country)Malaysia"
##  [9] "factor(country)Morocco"              "factor(country)Poland"
## [11] "factor(country)Saudi Arabia"         "factor(country)Sri Lanka"
## [13] "factor(country)Ukraine"              "factor(country)Uzbekistan"
```

```r
significativo_positivos
```

```
##  [1] "Xinternet"
##  [2] "Xplayers_ppl"
##  [3] "Xinflacion"
##  [4] "factor(country)Albania"
##  [5] "factor(country)Armenia"
##  [6] "factor(country)Australia"
##  [7] "factor(country)Austria"
##  [8] "factor(country)Azerbaijan"
##  [9] "factor(country)Bahrain"
## [10] "factor(country)Belarus"
## [11] "factor(country)Belgium"
## [12] "factor(country)Bolivia"
## [13] "factor(country)Bosnia and Herzegovina"
## [14] "factor(country)Bulgaria"
## [15] "factor(country)Cambodia"
## [16] "factor(country)Canada"
## [17] "factor(country)Chile"
## [18] "factor(country)Costa Rica"
## [19] "factor(country)Croatia"
## [20] "factor(country)Czech Republic"
## [21] "factor(country)Denmark"
```

```
## [22] "factor(country)Dominican Republic"
## [23] "factor(country)Ecuador"
## [24] "factor(country)Estonia"
## [25] "factor(country)Finland"
## [26] "factor(country)Georgia"
## [27] "factor(country)Greece"
## [28] "factor(country)Guatemala"
## [29] "factor(country)Hungary"
## [30] "factor(country)Iceland"
## [31] "factor(country)Ireland"
## [32] "factor(country)Israel"
## [33] "factor(country)Jordan"
## [34] "factor(country)Kazakhstan"
## [35] "factor(country)Kuwait"
## [36] "factor(country)Kyrgyzstan"
## [37] "factor(country)Lao People's Democratic Republic"
## [38] "factor(country)Latvia"
## [39] "factor(country)Lithuania"
## [40] "factor(country)Luxembourg"
## [41] "factor(country)Malta"
## [42] "factor(country)Moldova, Republic of"
## [43] "factor(country)Mongolia"
## [44] "factor(country)Netherlands"
## [45] "factor(country)New Zealand"
## [46] "factor(country)Nicaragua"
## [47] "factor(country)North Macedonia"
## [48] "factor(country)Norway"
## [49] "factor(country)Panama"
## [50] "factor(country)Paraguay"
## [51] "factor(country)Peru"
## [52] "factor(country)Portugal"
## [53] "factor(country)Romania"
## [54] "factor(country)Singapore"
## [55] "factor(country)Slovakia"
## [56] "factor(country)Slovenia"
## [57] "factor(country)Sweden"
## [58] "factor(country)Switzerland"
## [59] "factor(country)Trinidad and Tobago"
## [60] "factor(country)Tunisia"
## [61] "factor(country)Uruguay"
```

```
significativo_negativos
```

```
##  [1] "(Intercept)"
##  [2] "Xlife_exp"
##  [3] "Xdesempleo"
##  [4] "factor(country)Bangladesh"
##  [5] "factor(country)Brazil"
##  [6] "factor(country)China"
##  [7] "factor(country)Colombia"
##  [8] "factor(country)Egypt"
##  [9] "factor(country)France"
## [10] "factor(country)Germany"
## [11] "factor(country)India"
```

```
## [12] "factor(country)Indonesia"
## [13] "factor(country)Iran, Islamic Republic of"
## [14] "factor(country)Italy"
## [15] "factor(country)Japan"
## [16] "factor(country)Mexico"
## [17] "factor(country)Pakistan"
## [18] "factor(country)Philippines"
## [19] "factor(country)Russian Federation"
## [20] "factor(country)South Africa"
## [21] "factor(country)Thailand"
## [22] "factor(country)Turkey"
## [23] "factor(country)United Kingdom"
## [24] "factor(country)United States"
```

By year

- El año en sí mismo no parece tener un efecto significativo en Y después de ajustar por X

```
regresion_years = lm(Y~X+factor(year))
summary(regresion_years)
```

```
##
## Call:
## lm(formula = Y ~ X + factor(year))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4210 -0.6638  0.0667  0.8183  4.0918
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -13.136275   1.722143  -7.628 1.25e-13 ***
## Xpbicap           0.128044   0.149150   0.858   0.3910
## Xinternet         0.072457   0.007472   9.697  < 2e-16 ***
## Xlife_exp        -0.042880   0.027083  -1.583   0.1140
## Xpea             -5.870471   1.496042  -3.924 9.95e-05 ***
## Xdesempleo        0.002935   0.015434   0.190   0.8493
## Xpoblacion        7.025297   1.495512   4.698 3.42e-06 ***
## Xplayers_ppl      0.042720   0.004865   8.782  < 2e-16 ***
## Xinflacion       -0.016991   0.008222  -2.067   0.0393 *
## factor(year)2018  0.130596   0.221921   0.588   0.5565
## factor(year)2019  0.290034   0.227527   1.275   0.2030
## factor(year)2020 -0.152604   0.236746  -0.645   0.5195
## factor(year)2021  0.250853   0.243152   1.032   0.3027
## factor(year)2022  0.491172   0.290040   1.693   0.0910 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.489 on 490 degrees of freedom
##   (42 observations deleted due to missingness)
## Multiple R-squared:  0.6504, Adjusted R-squared:  0.6411
## F-statistic: 70.11 on 13 and 490 DF,  p-value: < 2.2e-16
```

OTRAS IDEAS

Top 3 country

```
df_panel %>%
  group_by(country) %>%
  top_n(3, total_earnings) %>%
  arrange(desc(total_earnings))
```

```
## # A tibble: 273 x 13
## # Groups:   country [91]
##    year  country   total_earnings total_players pbicap gdp_gr internet desempleo
##    <fct> <fct>              <dbl>         <int>  <dbl>  <dbl>    <dbl>     <dbl>
##  1 2021  China               17.8          2015   9.44   8.35     73.1      4.55
##  2 2022  China               17.7          2156   9.45   3.00     75.6      4.98
##  3 2019  United S~           17.7          6279  11.1    1.83     89.4      3.67
##  4 2022  United S~           17.3          4809  11.2    1.55     NA        3.65
##  5 2021  United S~           17.3          5154  11.2    5.78     91.8      5.35
##  6 2019  China               17.0          1455   9.22   5.58     64.1      4.56
##  7 2021  Russian ~           17.0          1119   9.44   5.53     88.2      4.72
##  8 2019  Korea, R~           16.7          1226  10.4    1.89     96.2      3.75
##  9 2018  Korea, R~           16.5          1283  10.4    2.46     96.0      3.82
## 10 2021  Korea, R~           16.4          1128  10.5    4.49     97.6      3.64
## # i 263 more rows
## # i 5 more variables: pea <dbl>, life_exp <dbl>, poblacion <dbl>,
## #   inflacion <dbl>, players_ppl <dbl>
```

Crecimiento de ganancia

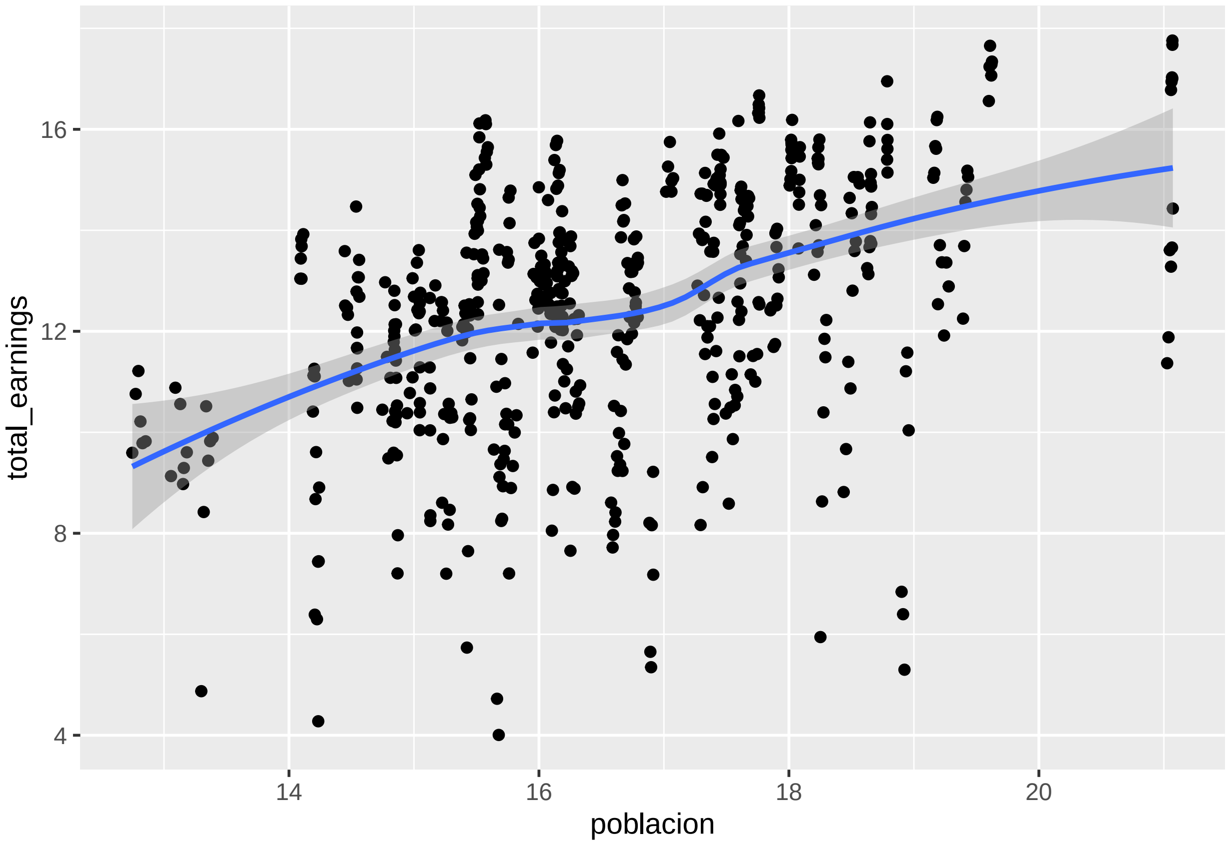
```
df_panel %>%
  group_by(country) %>%
  mutate(year = as.integer(year)) %>%
  summarise(CAGR = ifelse((last(year) - first(year)) != 0,
                         (last(total_earnings) / first(total_earnings))^(1/(last(year)-first(year))) -
  arrange(desc(CAGR))
```

```
## # A tibble: 91 x 2
##    country                            CAGR
##    <fct>                             <dbl>
##  1 Paraguay                         0.165
##  2 Iran, Islamic Republic of        0.155
##  3 Luxembourg                       0.152
##  4 Costa Rica                       0.132
##  5 Lao People's Democratic Republic 0.110
##  6 Bolivia                          0.0999
##  7 Egypt                            0.0904
##  8 Georgia                          0.0866
##  9 Azerbaijan                       0.0847
## 10 Bangladesh                       0.0797
## # i 81 more rows
```

Top 3 paises ganancias en el tiempo

```
ggplot(data = df_standar, aes(x = poblacion, y = total_earnings)) +
  geom_point() + # Añadimos los puntos del scatter plot
   geom_smooth() #method = "lm", se = TRUE, color = "blue") +
```

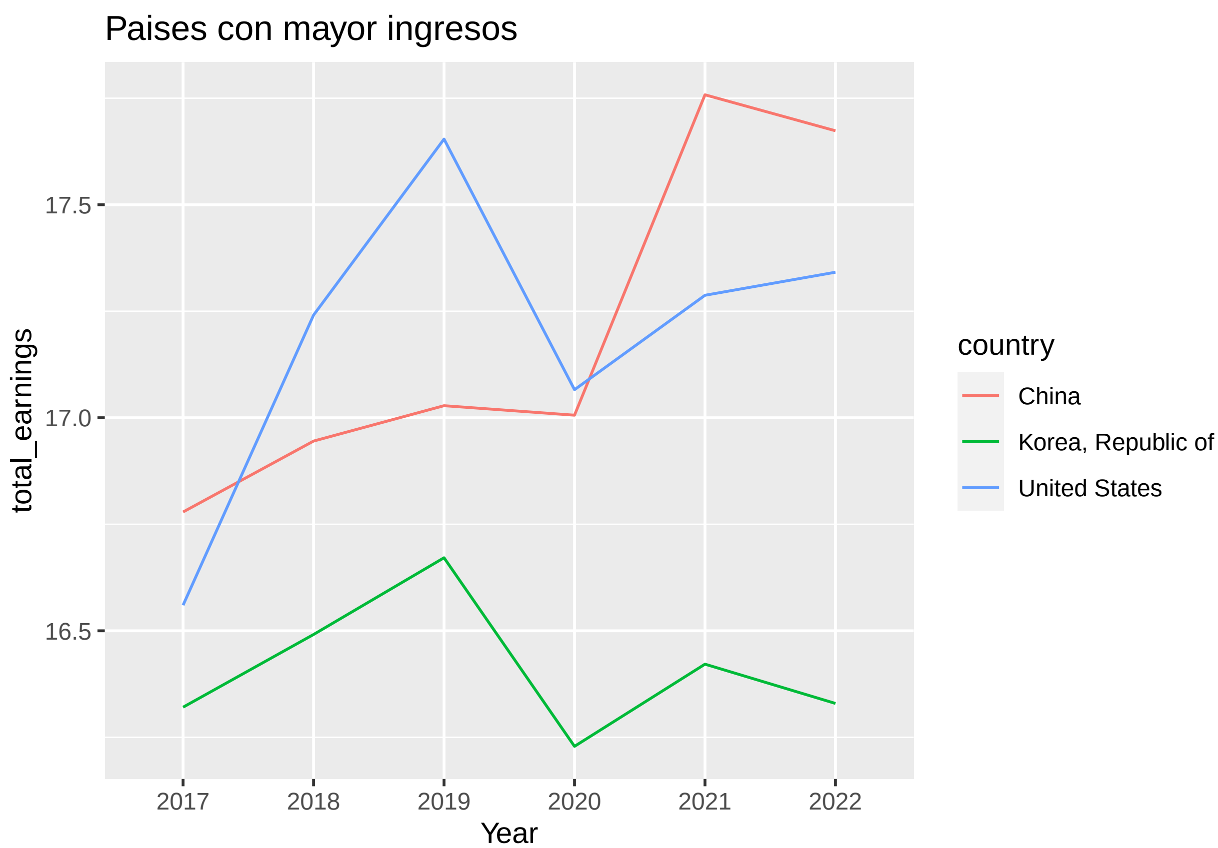
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
  labs(x = "Población", y = "Total de Ganancias", title = "Relación entre Población y Total de Ganancia
  theme_minimal() # Tema minimalista para el gráfico
```

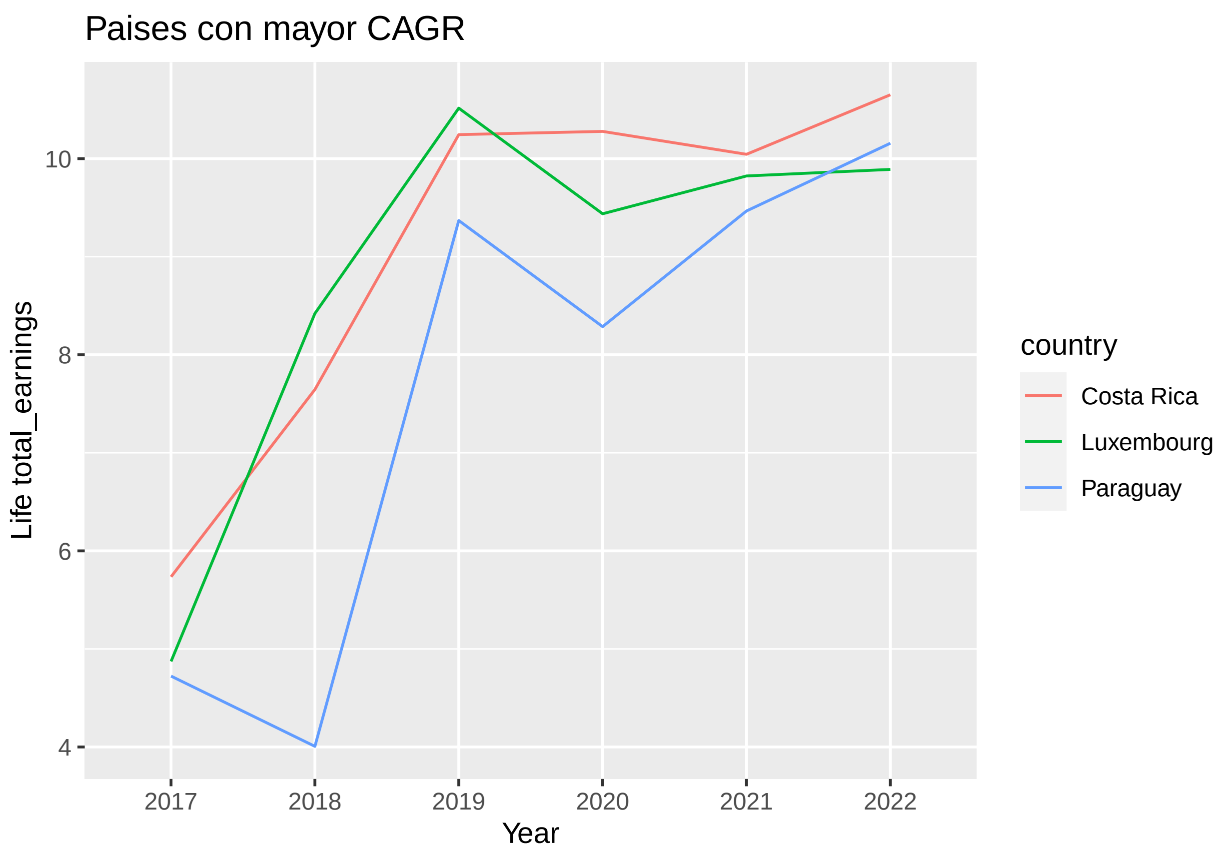
```
## NULL
```

```
ggplot(df_panel %>%
  group_by(year) %>%
  filter(country %in% c('China', 'United States', 'Korea, Republic of')),
  aes(x = year, y = total_earnings, group = country, color=country)) +
  geom_line() +
  labs(x = "Year", y = "total_earnings") +
  ggtitle("Paises con mayor ingresos")
```



Tops 3 paises crecimiento en el tiempo

```
ggplot(df_panel %>%
  group_by(year) %>%
  filter(country %in% c("Luxembourg", "Paraguay", "Costa Rica")),
  aes(x = year, y = total_earnings, group = country, color=country)) +
  geom_line() +
  labs(x = "Year", y = "Life total_earnings") +
  ggtitle("Paises con mayor CAGR")
```


Paises con mayor CAGR
Life total_earnings
10
8
6
4
2017
2018
2019
2020
2021
2022
Year
country
Costa Rica
Luxembourg
Paraguay