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,
         exp_tech, # tech access
         net_mig,
         life_exp, # edades
         poblacion# people
        ) %>%
  arrange(country, decreasing = FALSE)


head(df, 3)
```

```
##         year country total_earnings total_players   pbicap   gdp_gr CPI
## 1 2017-01-01 Albania        2868.16             2 4531.032 3.898112  38
## 2 2018-01-01 Albania        1346.55             3 5287.661 4.276312  36
## 3 2019-01-01 Albania       37459.64            14 5396.214 2.523541  35
##   internet elect_acc exp_tech net_mig life_exp poblacion
## 1 62.40000     99.89 31.52551   -9768   79.047   2873457
## 2 65.40000    100.00 30.75579   -9106   79.184   2866376
## 3 68.55039    100.00 31.18889   -8889   79.282   2854191
```

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          CPI     internet    elect_acc     exp_tech 
##             8            6           54           99           38 
##       net_mig     life_exp    poblacion 
##            12           99            0
```

- 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, ]


# 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
```

```r
### 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.na(x)))
```

```
##           year        country total_earnings  total_players         pbicap
##              0              0              0              0              0
##         gdp_gr            CPI       internet      elect_acc       exp_tech
##              0              0              0              0              0
##        net_mig       life_exp      poblacion    players_ppl
##              0              0              0              0
```

3. Normalizacion con logaritmo

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

```r
df_standar <- df %>%
  mutate(across(c("total_earnings", "pbicap", "poblacion"), ~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

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

```
## [1] 90  5
```

```r
colnames(df_standar)
```

```
##  [1] "year"           "country"        "total_earnings" "total_players"
##  [5] "pbicap"         "gdp_gr"         "CPI"            "internet"
##  [9] "elect_acc"      "exp_tech"       "net_mig"        "life_exp"
## [13] "poblacion"      "players_ppl"
```

- definimos las variables para el modelo

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

```r
           poblacion,
           players_ppl,
           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 CPI
## Albania-2017 2017 Albania       7.961426             2 8.418705 3.898112  38
## Albania-2018 2018 Albania       7.205301             3 8.573131 4.276312  36
## Albania-2019 2019 Albania      10.531019            14 8.593453 2.523541  35
##              internet elect_acc exp_tech   net_mig life_exp poblacion
## Albania-2017 62.40000     99.89 31.52551 -3399.390   79.047  14.87103
## Albania-2018 65.40000    100.00 30.75579 -3176.834   79.184  14.86856
## Albania-2019 68.55039    100.00 31.18889 -3114.368   79.282  14.86430
##              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"))
##
## Balanced Panel: n = 90, T = 5, N = 450
##
## Residuals:
##        Min.     1st Qu.      Median     3rd Qu.        Max.
## -3.08315004 -0.39197841 -0.00066738  0.46132642  2.60355151
##
## Coefficients:
##                Estimate  Std. Error t-value  Pr(>|t|)
## Xpbicap       8.4934e-01  6.3485e-01  1.3378 0.1818084
## Xgdp_gr       1.6811e-02  1.0810e-02  1.5551 0.1208117
## Xinternet     4.5395e-02  1.0050e-02  4.5171 8.559e-06 ***
## Xlife_exp    -9.0847e-02  5.2311e-02 -1.7367 0.0833162 .
## Xpoblacion    9.3142e+00  3.2383e+00  2.8763 0.0042686 **
## Xplayers_ppl  2.7940e-02  8.0547e-03  3.4688 0.0005874 ***
## Xnet_mig      6.8660e-06  8.8396e-06  0.7767 0.4378366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Total Sum of Squares:    358.88
## Residual Sum of Squares: 253.18
## R-Squared:      0.29454
## Adj. R-Squared: 0.10269
## F-statistic: 21.0546 on 7 and 353 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"))
##
## Balanced Panel: n = 90, T = 5, N = 450
##
## Effects:
##                  var std.dev share
## idiosyncratic 0.7172  0.8469 0.329
## individual    1.4619  1.2091 0.671
## theta: 0.7011
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -3.901878 -0.352216  0.061909  0.496691  2.301204
##
## Coefficients:
##                 Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)  -1.1228e+01  2.6684e+00 -4.2079 2.577e-05 ***
## Xpbicap       3.2064e-01  2.1563e-01  1.4870   0.13701
## Xgdp_gr       2.3049e-02  9.6591e-03  2.3862   0.01702 *
## Xinternet     6.3383e-02  6.6849e-03  9.4815 < 2.2e-16 ***
## Xlife_exp    -4.7840e-02  3.6891e-02 -1.2968   0.19470
## Xpoblacion    1.1511e+00  8.9533e-02 12.8570 < 2.2e-16 ***
## Xplayers_ppl  3.4609e-02  6.3188e-03  5.4771 4.324e-08 ***
## Xnet_mig     -2.5391e-06  8.4595e-06 -0.3002   0.76406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    578.89
## Residual Sum of Squares: 326.84
## R-Squared:      0.4354
## Adj. R-Squared: 0.42645
## Chisq: 340.85 on 7 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"))
##
## Balanced Panel: n = 90, T = 5, N = 450
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -6.32915 -0.58996  0.15217  0.80292  4.45191
##
## Coefficients:
##                 Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)  -1.2868e+01  1.6582e+00 -7.7603 5.927e-14 ***
## Xpbicap       1.5554e-01  1.4852e-01  1.0473  0.295541
## Xgdp_gr       4.6087e-02  1.5065e-02  3.0592  0.002354 **
## Xinternet     6.7141e-02  6.4625e-03 10.3892 < 2.2e-16 ***
## Xlife_exp    -2.1521e-02  2.7495e-02 -0.7827  0.434207
## Xpoblacion    1.1955e+00  4.9085e-02 24.3553 < 2.2e-16 ***
## Xplayers_ppl  4.7642e-02  4.9791e-03  9.5684 < 2.2e-16 ***
## Xnet_mig     -3.5739e-05  1.1821e-05 -3.0234  0.002645 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    2821
## Residual Sum of Squares: 979.76
## R-Squared:      0.65269
## Adj. R-Squared: 0.64719
## F-statistic: 118.664 on 7 and 442 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

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

```
##
##  Lagrange Multiplier Test - (Breusch-Pagan)
##
## data:  Y ~ X
## chisq = 367.19, 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 = 29.334, df = 7, p-value = 0.0001257
## 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 = 11.383, df1 = 89, df2 = 353, p-value < 2.2e-16
## alternative hypothesis: significant effects
```

Regresiones

Regresieon con efectos fijos

by Country Spain

```
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

```
no_significativo
```

```
##  [1] "Xpbicap"                           "Xgdp_gr"
##  [3] "Xlife_exp"                         "Xnet_mig"
##  [5] "factor(country)Algeria"            "factor(country)Argentina"
##  [7] "factor(country)Colombia"           "factor(country)France"
##  [9] "factor(country)Iraq"               "factor(country)Italy"
## [11] "factor(country)Korea, Republic of" "factor(country)Morocco"
## [13] "factor(country)Philippines"        "factor(country)Saudi Arabia"
## [15] "factor(country)South Africa"       "factor(country)Sri Lanka"
## [17] "factor(country)Thailand"           "factor(country)Ukraine"
## [19] "factor(country)Uzbekistan"
```

```
significativo_positivos
```

```
##  [1] "Xinternet"
##  [2] "Xpoblacion"
##  [3] "Xplayers_ppl"
##  [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)Malaysia"
## [42] "factor(country)Malta"
## [43] "factor(country)Moldova, Republic of"
## [44] "factor(country)Mongolia"
## [45] "factor(country)Netherlands"
## [46] "factor(country)New Zealand"
## [47] "factor(country)Nicaragua"
## [48] "factor(country)North Macedonia"
## [49] "factor(country)Norway"
## [50] "factor(country)Panama"
## [51] "factor(country)Paraguay"
## [52] "factor(country)Peru"
## [53] "factor(country)Poland"
## [54] "factor(country)Portugal"
## [55] "factor(country)Romania"
## [56] "factor(country)Singapore"
## [57] "factor(country)Slovakia"
## [58] "factor(country)Slovenia"
## [59] "factor(country)Sweden"
## [60] "factor(country)Switzerland"
## [61] "factor(country)Trinidad and Tobago"
## [62] "factor(country)Tunisia"
## [63] "factor(country)Uruguay"
```

```
significativo_negativos
```

```
##  [1] "(Intercept)"                          "factor(country)Bangladesh"
##  [3] "factor(country)Brazil"                "factor(country)China"
##  [5] "factor(country)Egypt"                 "factor(country)Germany"
##  [7] "factor(country)India"                 "factor(country)Indonesia"
##  [9] "factor(country)Japan"                 "factor(country)Mexico"
## [11] "factor(country)Pakistan"              "factor(country)Russian Federation"
## [13] "factor(country)Turkey"                "factor(country)United Kingdom"
## [15] "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
## -6.5894 -0.6268  0.1054  0.7651  4.6485
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.288e+01  1.698e+00  -7.584 2.03e-13 ***
## Xpbicap           1.832e-01  1.499e-01   1.222  0.22244
## Xgdp_gr           6.474e-02  2.435e-02   2.658  0.00814 **
## Xinternet         6.643e-02  7.058e-03   9.411  < 2e-16 ***
## Xlife_exp        -2.558e-02  2.835e-02  -0.902  0.36737
## Xpoblacion        1.191e+00  4.962e-02  23.999  < 2e-16 ***
## Xplayers_ppl      4.645e-02  5.069e-03   9.165  < 2e-16 ***
## Xnet_mig         -3.281e-05  1.218e-05  -2.693  0.00736 **
## factor(year)2018  1.463e-01  2.230e-01   0.656  0.51204
## factor(year)2019  3.406e-01  2.282e-01   1.492  0.13634
## factor(year)2020  3.520e-01  3.026e-01   1.163  0.24541
## factor(year)2021  1.634e-02  2.573e-01   0.063  0.94941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.491 on 438 degrees of freedom
## Multiple R-squared:  0.6551, Adjusted R-squared:  0.6464
## F-statistic: 75.61 on 11 and 438 DF,  p-value: < 2.2e-16
```

OTRAS IDEAS

Top 3 country

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

```
## # A tibble: 270 x 14
## # Groups:   country [90]
##    year  country        total_earnings total_players pbicap gdp_gr   CPI internet
##    <fct> <fct>                   <dbl>         <int>  <dbl>  <dbl> <dbl>    <dbl>
##  1 2021  China                    17.8          2015   9.44   8.35    45     73.1
##  2 2019  United States            17.7          6279  11.1    1.83    69     89.4
##  3 2021  United States            17.3          5132  11.2    5.78    67     91.8
##  4 2018  United States            17.2          4413  11.0    2.40    71     88.5
##  5 2019  China                    17.0          1455   9.22   5.58    41     64.1
##  6 2020  China                    17.0          1614   9.25   2.00    42     70.1
##  7 2021  Russian Fede~            17.0          1112   9.44   5.53    29     88.2
##  8 2019  Korea, Repub~            16.7          1226  10.4    1.89    59     96.2
##  9 2018  Korea, Repub~            16.5          1283  10.4    2.46    57     96.0
## 10 2021  Korea, Repub~            16.4          1129  10.5    4.49    62     97.6
## # i 260 more rows
## # i 6 more variables: elect_acc <dbl>, exp_tech <dbl>, net_mig <dbl>,
## #   life_exp <dbl>, poblacion <dbl>, players_ppl <dbl>
```

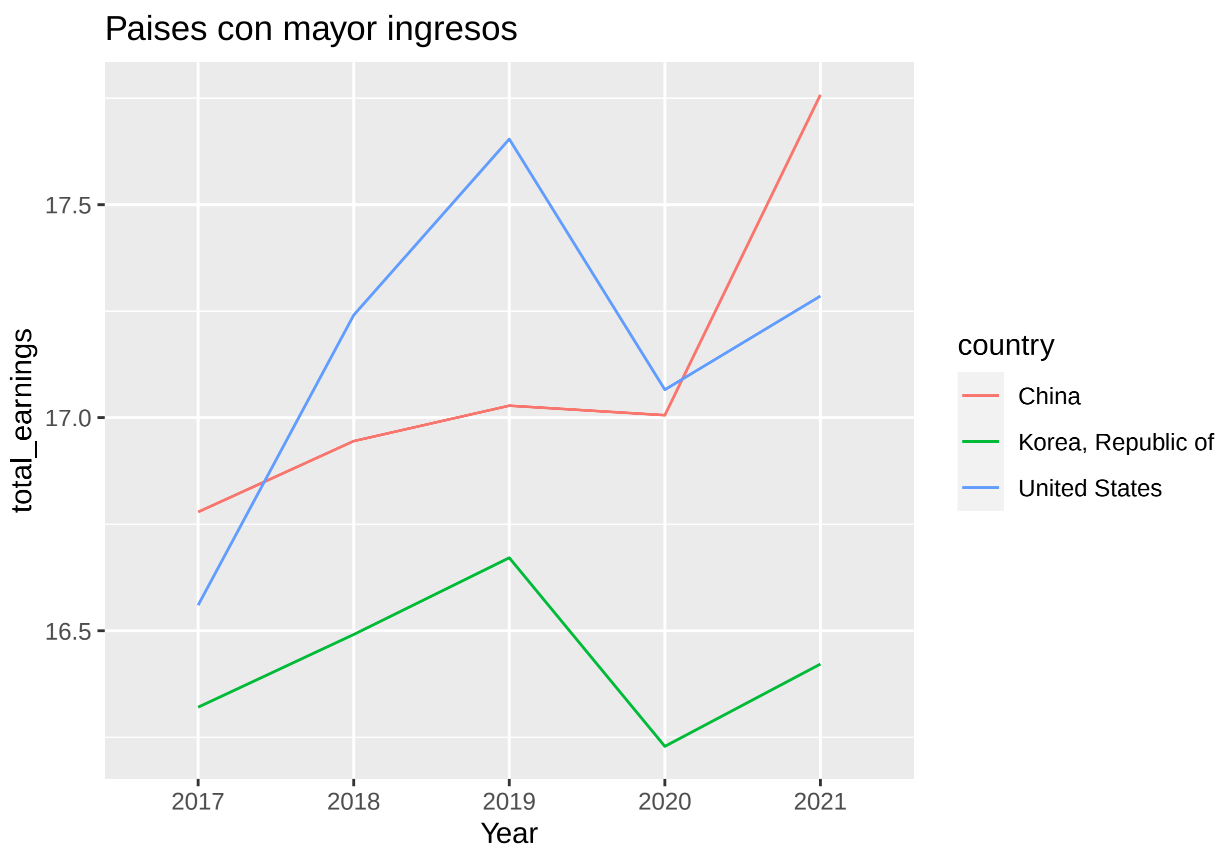
Crecimiento de ganancia

```
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: 90 x 2
##    country                            CAGR
##    <fct>                             <dbl>
##  1 Luxembourg                       0.192
##  2 Paraguay                         0.190
##  3 Costa Rica                       0.150
##  4 Bangladesh                       0.141
##  5 Bolivia                          0.125
##  6 Azerbaijan                       0.113
##  7 Uzbekistan                       0.103
##  8 Egypt                            0.0978
##  9 Lao People's Democratic Republic 0.0945
## 10 Georgia                          0.0780
## # i 80 more rows
```

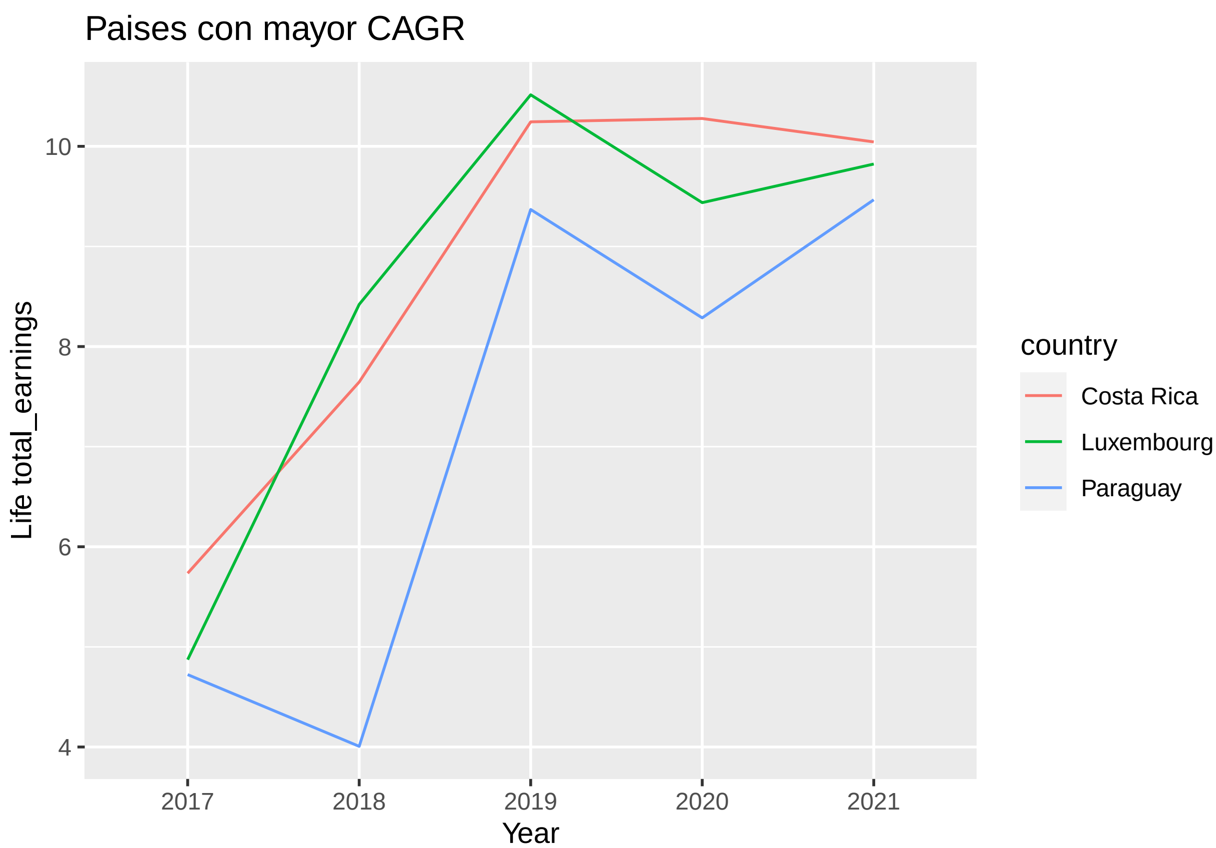
Top 3 paises ganancias en el tiempo

```
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
Year
country
Costa Rica
Luxembourg
Paraguay