modelo econometrico version 4

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

# df <- df %>%
#   filter(year < as.Date("2022-01-01"))

## "Cuba"  "Lebanon"  "Syrian Arab Republic"  "Venezuela"
pbicap_faltantes <- unique(df[is.na(df$pbicap), ]$country)
desemp_faltantes <- unique(df[is.na(df$desempleo), ]$country)
internet_faltantes <- unique(df[is.na(df$internet), ]$country)
# netmig_faltantes <- unique(df[is.na(df$net_mig), ]$country)
df <- df[!df$country %in% pbicap_faltantes, ]
df <- df[!df$country %in% desemp_faltantes, ]
df <- df[!df$country %in% internet_faltantes, ]
# df <- df[!df$country %in% netmig_faltantes, ]

###########

### Jugadores por poblacion por millon
df$players_ppl <- (df$total_players/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       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

```r
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

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

```
## [1] 52  6
```

```r
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

```r
attach(df_standar)
Y <- cbind(total_earnings)
X <- cbind(pbicap,
           # gdp_gr,
           internet,
           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"))
##
## Balanced Panel: n = 52, T = 6, N = 312
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -3.027154 -0.302260 -0.020244  0.302458  1.936583
##
## Coefficients:
##               Estimate Std. Error t-value  Pr(>|t|)
## Xpbicap       1.0203501  0.5469043  1.8657   0.06325 .
## Xinternet     0.0528553  0.0094360  5.6014 5.554e-08 ***
## Xlife_exp    -0.0622869  0.0538426 -1.1568   0.24844
## Xpea         14.5168032  6.0728331  2.3904   0.01756 *
## Xdesempleo   -0.0513053  0.0407549 -1.2589   0.20924
## Xpoblacion    0.3235078  7.3895645  0.0438   0.96512
## Xplayers_ppl  0.0343217  0.0071491  4.8008 2.711e-06 ***
## Xinflacion    0.0014463  0.0066226  0.2184   0.82730
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    220.37
## Residual Sum of Squares: 121.95
## R-Squared:      0.44664
## Adj. R-Squared: 0.31708
## F-statistic: 25.4247 on 8 and 252 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 = 52, T = 6, N = 312
##
## Effects:
##                   var std.dev share
```

```
## idiosyncratic 0.4839  0.6956 0.303
## individual    1.1126  1.0548 0.697
## theta: 0.74
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -4.017165 -0.342340  0.014746  0.393891  1.977755
##
## Coefficients:
##                  Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)  -13.7337682   3.9624281 -3.4660 0.0005283 ***
## Xpbicap        0.1475137   0.2783104  0.5300 0.5960891
## Xinternet      0.0715762   0.0081105  8.8251 < 2.2e-16 ***
## Xlife_exp     -0.0145074   0.0441969 -0.3282 0.7427261
## Xpea          -0.5220489   2.6526334 -0.1968 0.8439809
## Xdesempleo    -0.0103262   0.0298161 -0.3463 0.7290953
## Xpoblacion     1.7331544   2.6602213  0.6515 0.5147189
## Xplayers_ppl   0.0328438   0.0063247  5.1930  2.07e-07 ***
## Xinflacion     0.0031747   0.0065537  0.4844 0.6280972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    307.39
## Residual Sum of Squares: 165.77
## R-Squared:      0.46071
## Adj. R-Squared: 0.44647
## Chisq: 258.847 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"))
##
## Balanced Panel: n = 52, T = 6, N = 312
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -6.344728 -0.447889  0.062565  0.585752  3.890074
##
## Coefficients:
##                  Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)  -16.8999987   2.1040994 -8.0319 2.149e-14 ***
## Xpbicap       -0.1972268   0.1892231 -1.0423 0.2981047
## Xinternet      0.0752989   0.0090421  8.3276 2.880e-15 ***
## Xlife_exp      0.0308779   0.0328703  0.9394 0.3482806
## Xpea          -5.2996778   1.4498795 -3.6553 0.0003028 ***
```

```
## Xdesempleo    -0.0301691   0.0201566 -1.4967 0.1355040
## Xpoblacion     6.5543363   1.4518762  4.5144 9.102e-06 ***
## Xplayers_ppl   0.0449828   0.0050021  8.9928 < 2.2e-16 ***
## Xinflacion    -0.0060329   0.0077030 -0.7832 0.4341294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    1507.8
## Residual Sum of Squares: 460.39
## R-Squared:      0.69466
## Adj. R-Squared: 0.6866
## F-statistic: 86.1688 on 8 and 303 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 = 287.65, 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

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

```
##
##  Hausman Test
##
## data:  Y ~ X
## chisq = 48.764, df = 8, p-value = 7.054e-08
## 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

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

```
##
##  F test for individual effects
##
## data:  Y ~ X
## F = 13.714, df1 = 51, df2 = 252, 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"                     "Xlife_exp"
## [3] "Xdesempleo"                  "Xpoblacion"
## [5] "Xinflacion"                  "factor(country)Argentina"
## [7] "factor(country)Colombia"     "factor(country)Saudi Arabia"
```

```r
significativo_positivos
```

```
##  [1] "Xinternet"
##  [2] "Xpea"
##  [3] "Xplayers_ppl"
##  [4] "factor(country)Albania"
##  [5] "factor(country)Austria"
##  [6] "factor(country)Belarus"
```

```
##  [7] "factor(country)Belgium"
##  [8] "factor(country)Bosnia and Herzegovina"
##  [9] "factor(country)Bulgaria"
## [10] "factor(country)Costa Rica"
## [11] "factor(country)Croatia"
## [12] "factor(country)Czech Republic"
## [13] "factor(country)Denmark"
## [14] "factor(country)Ecuador"
## [15] "factor(country)Estonia"
## [16] "factor(country)Finland"
## [17] "factor(country)Georgia"
## [18] "factor(country)Greece"
## [19] "factor(country)Hong Kong"
## [20] "factor(country)Hungary"
## [21] "factor(country)Iraq"
## [22] "factor(country)Kazakhstan"
## [23] "factor(country)Latvia"
## [24] "factor(country)Lithuania"
## [25] "factor(country)Luxembourg"
## [26] "factor(country)Malaysia"
## [27] "factor(country)Malta"
## [28] "factor(country)Netherlands"
## [29] "factor(country)Norway"
## [30] "factor(country)Paraguay"
## [31] "factor(country)Peru"
## [32] "factor(country)Poland"
## [33] "factor(country)Portugal"
## [34] "factor(country)Romania"
## [35] "factor(country)Singapore"
## [36] "factor(country)Slovenia"
## [37] "factor(country)Sweden"
## [38] "factor(country)United Arab Emirates"
## [39] "factor(country)Uruguay"
```

```
significativo_negativos
```

```
##  [1] "(Intercept)"                        "factor(country)Brazil"
##  [3] "factor(country)China"               "factor(country)Egypt"
##  [5] "factor(country)France"              "factor(country)Germany"
##  [7] "factor(country)Indonesia"           "factor(country)Italy"
##  [9] "factor(country)Korea, Republic of" "factor(country)Russian Federation"
## [11] "factor(country)Thailand"            "factor(country)Turkey"
## [13] "factor(country)Viet Nam"
```

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
## -6.3486 -0.4850  0.0561  0.5701  4.0638
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -16.620623   2.106497  -7.890 5.79e-14 ***
## Xpbicap           -0.158490   0.192749  -0.822 0.411586
## Xinternet          0.073347   0.010364   7.077 1.06e-11 ***
## Xlife_exp          0.023061   0.033429   0.690 0.490823
## Xpea              -4.990160   1.482923  -3.365 0.000865 ***
## Xdesempleo        -0.022897   0.020378  -1.124 0.262076
## Xpoblacion         6.245015   1.486041   4.202 3.49e-05 ***
## Xplayers_ppl       0.045299   0.005003   9.054  < 2e-16 ***
## Xinflacion        -0.008500   0.008022  -1.060 0.290165
## factor(year)2018   0.180484   0.241844   0.746 0.456084
## factor(year)2019   0.541695   0.247702   2.187 0.029528 *
## factor(year)2020  -0.102035   0.259212  -0.394 0.694133
## factor(year)2021   0.229407   0.266492   0.861 0.390019
## factor(year)2022   0.304390   0.274964   1.107 0.269180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.225 on 298 degrees of freedom
## Multiple R-squared:  0.7033, Adjusted R-squared:  0.6904
## F-statistic: 54.34 on 13 and 298 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: 156 x 13
## # Groups:   country [52]
##    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  China               17.0          1455   9.22   5.58     64.1      4.56
##  4 2021  Russian ~           17.0          1119   9.44   5.53     88.2      4.72
##  5 2019  Korea, R~           16.7          1226  10.4    1.89     96.2      3.75
##  6 2018  Korea, R~           16.5          1283  10.4    2.46     96.0      3.82
##  7 2021  Korea, R~           16.4          1128  10.5    4.49     97.6      3.64
##  8 2022  Brazil              16.2          1573   9.10   2.43     80.5      9.23
```

```
##   9 2019  France              16.2          1340   10.6    1.50     83.3      8.41
##  10 2021  Brazil              16.2          1446    8.95   4.44     80.7     13.2
## # i 146 more rows
## # i 5 more variables: pea <dbl>, life_exp <dbl>, poblacion <dbl>,
## #   inflacion <dbl>, players_ppl <dbl>
```

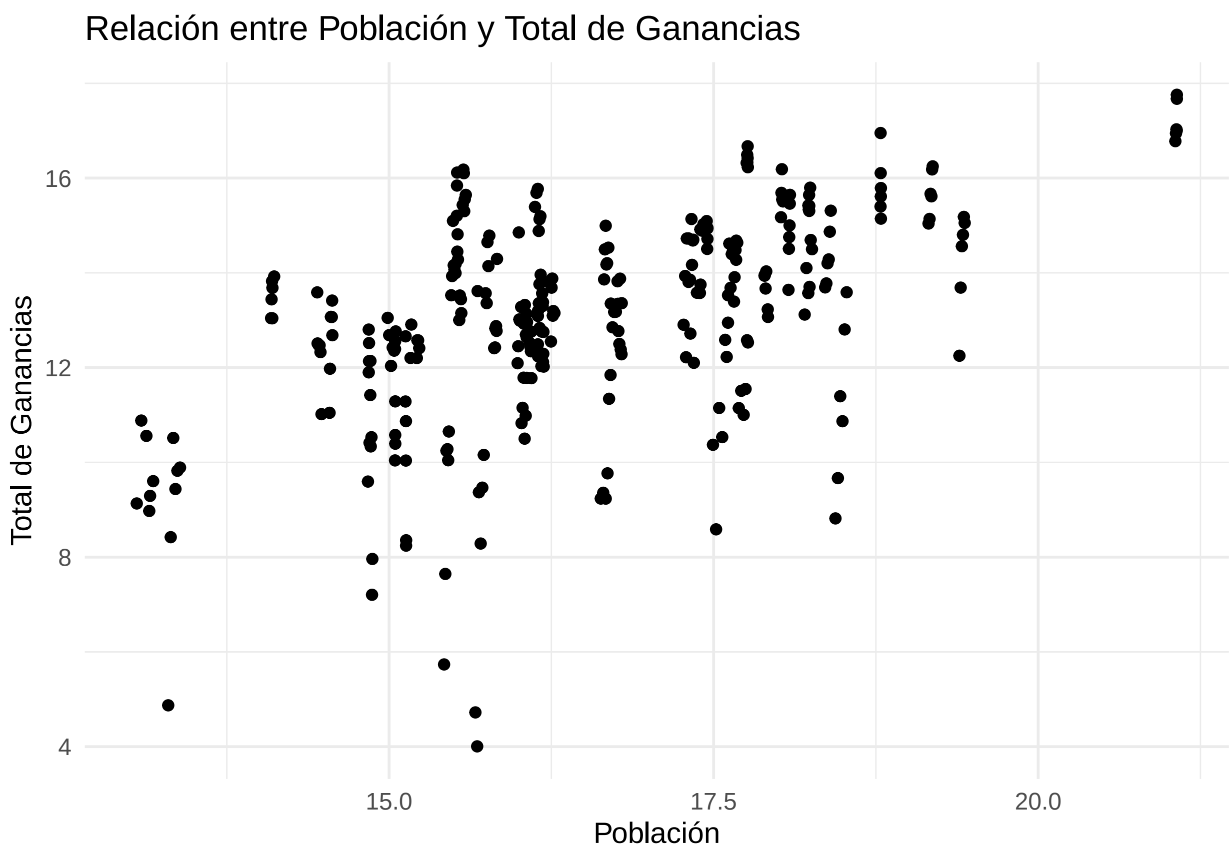
Crecimiento de ganancia

```r
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))
```

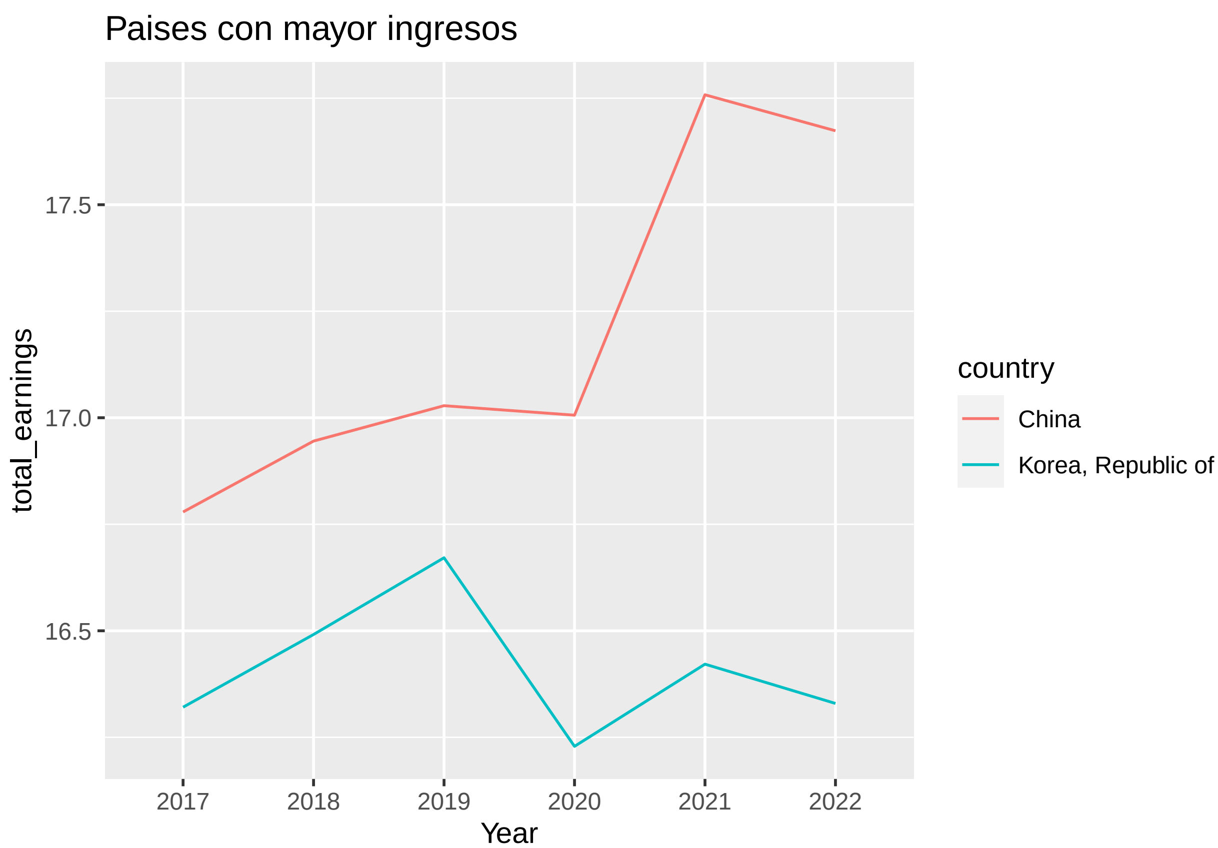
```
## # A tibble: 52 x 2
##    country        CAGR
##    <fct>         <dbl>
##  1 Paraguay     0.165
##  2 Luxembourg   0.152
##  3 Costa Rica   0.132
##  4 Egypt        0.0904
##  5 Georgia      0.0866
##  6 Iraq         0.0546
##  7 Ecuador      0.0510
##  8 Uruguay      0.0430
##  9 Latvia       0.0428
## 10 Saudi Arabia 0.0424
## # i 42 more rows
```

Top 3 paises ganancias en el tiempo

```r
ggplot(data = df_panel, aes(x = poblacion, y = total_earnings)) +
  geom_point() + # Añadimos los puntos del scatter plot
   #geom_smooth() #method = "lm", se = TRUE, color = "blue") +
  labs(x = "Población", y = "Total de Ganancias", title = "Relación entre Población y Total de Ganancias
  theme_minimal() # Tema minimalista para el gráfico
```

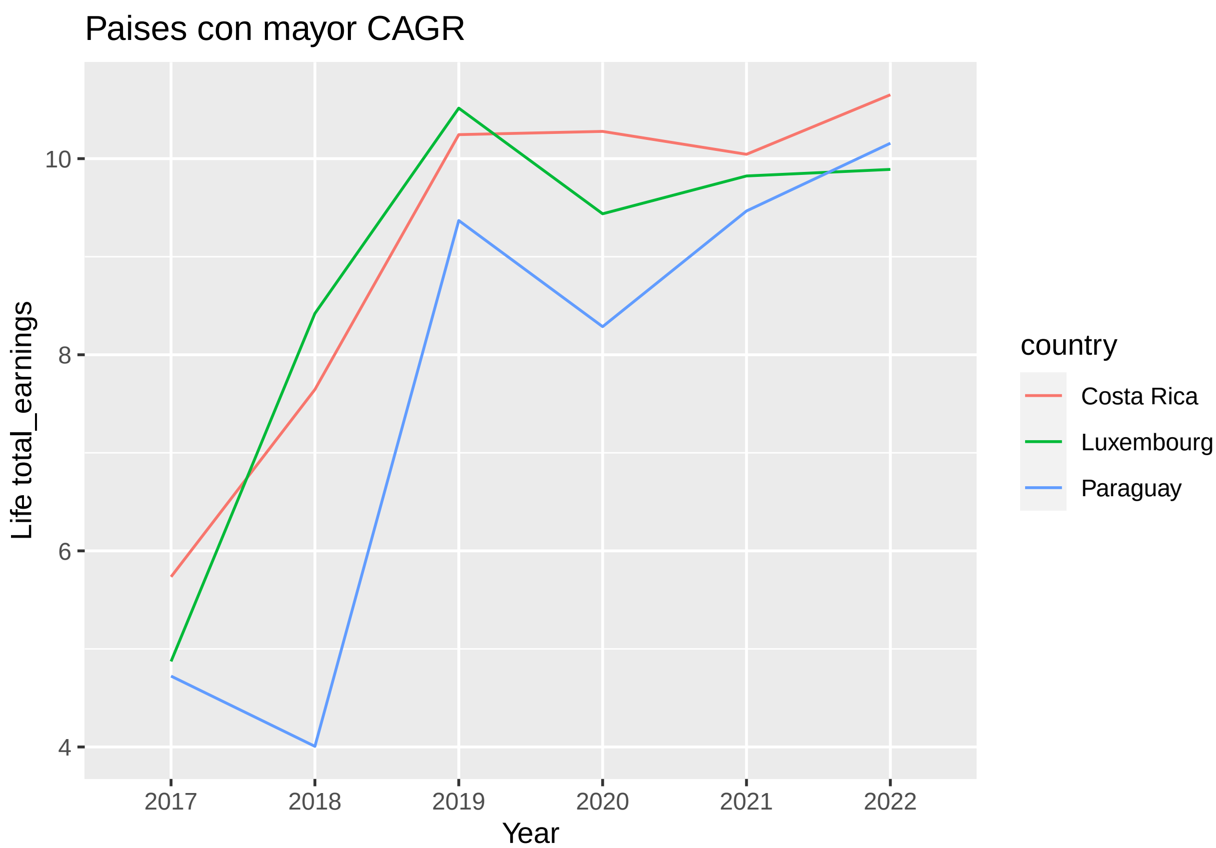



```r
ggplot(df_panel %>%
  group_by(year) %>%
  filter(country %in% c('China', 'United States', 'Korea, Republic of')),
  aes(x = year, y = total_earnings, group = country, color=country)) +
  geom_line() +
  labs(x = "Year", y = "total_earnings") +
  ggtitle("Paises con mayor ingresos")
```



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