

Documento_jupyter

June 24, 2024

1 Replicación de “Global inequality remotely sensed”

Ignacio Sarmiento-Barbieri

1.1 Introduction

Over the past decades, there has been a transition from theoretical to data-driven inequality research (1). However, progress is limited by a lack of data on economic prosperity at the household level (2) as well as the absence of consensus on ways of measuring economic inequality (3, 4). Practical constraints include limited coverage, incomparability at population subscales, dependence on misreported income surveys, and low-quality data collection in developing economies (5–7).

```
[16]: if(!require(pacman)){install.packages('pacman');require(pacman)}

# Instalar paquetes
cat(version$version.string)
paquetes <- c(
  # Básicos
  "here",
  "tidyverse",
  "rio",
  "janitor",
  # Econometría
  "fixest",
  "lme4",
  "modelsummary",
  "visreg",
  "stargazer",
  # Modelos Estadísticos
  "MuMIn",
  # Diagramas
  "visreg",
  "ggplot2",
  "RColorBrewer",
  "kableExtra",
  # Espacial
  "sp"
)
```

```
# Instalar paquetes usando librería "pacman" -----
if(!require(pacman)){install.packages(pacman); library(pacman)}
p_load(paquetes, character.only = T)
```

R version 4.3.1 (2023-06-16)

Para cargar los datos si se está trabajando desde jupyter localmente (ejecutándose en la propia computadora) cargar el archivo `.RData` se hace directamente con la función `load`:

```
load(here("../02_analysis/01_input", 'pnas_light_ineq.RData'))
```

Pero si estás trabajando en línea en *Google Colab*, *Renku* o *Binder* la solución más sencilla sería manualmente descargar `pnas_light_ineq.RData` y subirlo al ambiente de desarrollo. Lo adecuado sería automatizar la adquisición de los datos, como se hizo en la demostración en vivo durante la presentación.

```
[2]: load(here("../02_analysis/01_input", 'pnas_light_ineq.RData'))
```

```
Error in here("../02_analysis/01_input", "pnas_light_ineq.RData"): impossible d
↳ trouver la fonction "here"
Traceback:
1. load(here("../02_analysis/01_input", "pnas_light_ineq.RData"))
```

2 Replicar Tabla 1

Note que al inicio de la siguiente celda incluí el comando magic ([magic commands](#)) `%%capture cap --no-stderr`. Esto cumple una función equivalente a `results = 'hide'` en nuestro documento `Documento.Rmd` de R Markdown.

Note también que ejecutar la celda desde Jupyter/VS Code/Google Colab con esta línea **no funciona**. Mientras se esté haciendo análisis exploratorio esta línea debe mantenerse comentada, pero al exportar el documento a algún tipo de insumo se debe “activar” o “descomentar”.

```
[1]: %%capture cap --no-stderr

mod_1 <- lmer(Gini ~ Light_Gini +
              (1 | Year),
              data = df_nat)
summary(mod_1)
r.squaredGLMM(mod_1)[2]
AIC(mod_1)
AICc(mod_1)

mod_2 <- lmer(Gini ~ Light_Gini +
              log(POP) +
              (1 | Year),
```

```

      data = df_nat)
summary(mod_2)
r.squaredGLMM(mod_2)[2]
AIC(mod_2)
AICc(mod_2)

mod_3 <- lmer(Gini ~ Light_Gini +
              log(POP) +
              log(GDP) +
              (1 | Year),
              data = df_nat)
summary(mod_3)
r.squaredGLMM(mod_3)[2]
AIC(mod_3)
AICc(mod_3)

mod_4 <- lm(Gini ~ Light_Gini +
            log(POP) +
            log(GDP),
            data = df_nat)
summary(mod_4)

```

```

Error in parse(text = x, srcfile = src): <text>:1:1: SPECIAL inattendu(e)
1: %%
  ~
Traceback:

```

Ahora veamos los resultados. El paquete `modelsummary` presentó problemas pues en R Markdown la opción `results = 'asis'` del chunk de código permite que al compilar, el código producido por la función se interprete como \LaTeX , pero en este caso logré corregirlo, entonces imprimí los modelos usando el paquete `stargazer`

```

[18]: #modelsummary(list(mod_1,mod_2,mod_3,mod_4))
      stargazer(list(mod_1,mod_2,mod_3,mod_4), type = 'text')

```

```

=====
                        Dependent variable:
-----
                                Gini
                linear          OLS
            mixed-effects
            (1)      (2)      (3)      (4)
-----
Light_Gini      0.334***  0.290***  0.201***  0.171***

```

	(0.041)	(0.041)	(0.027)	(0.028)
log(POP)		1.561*** (0.358)	6.648*** (0.389)	6.830*** (0.415)
log(GDP)			-5.704*** (0.352)	-5.884*** (0.374)
Constant	15.657*** (2.657)	-7.817 (5.956)	63.156*** (5.853)	66.889*** (6.108)

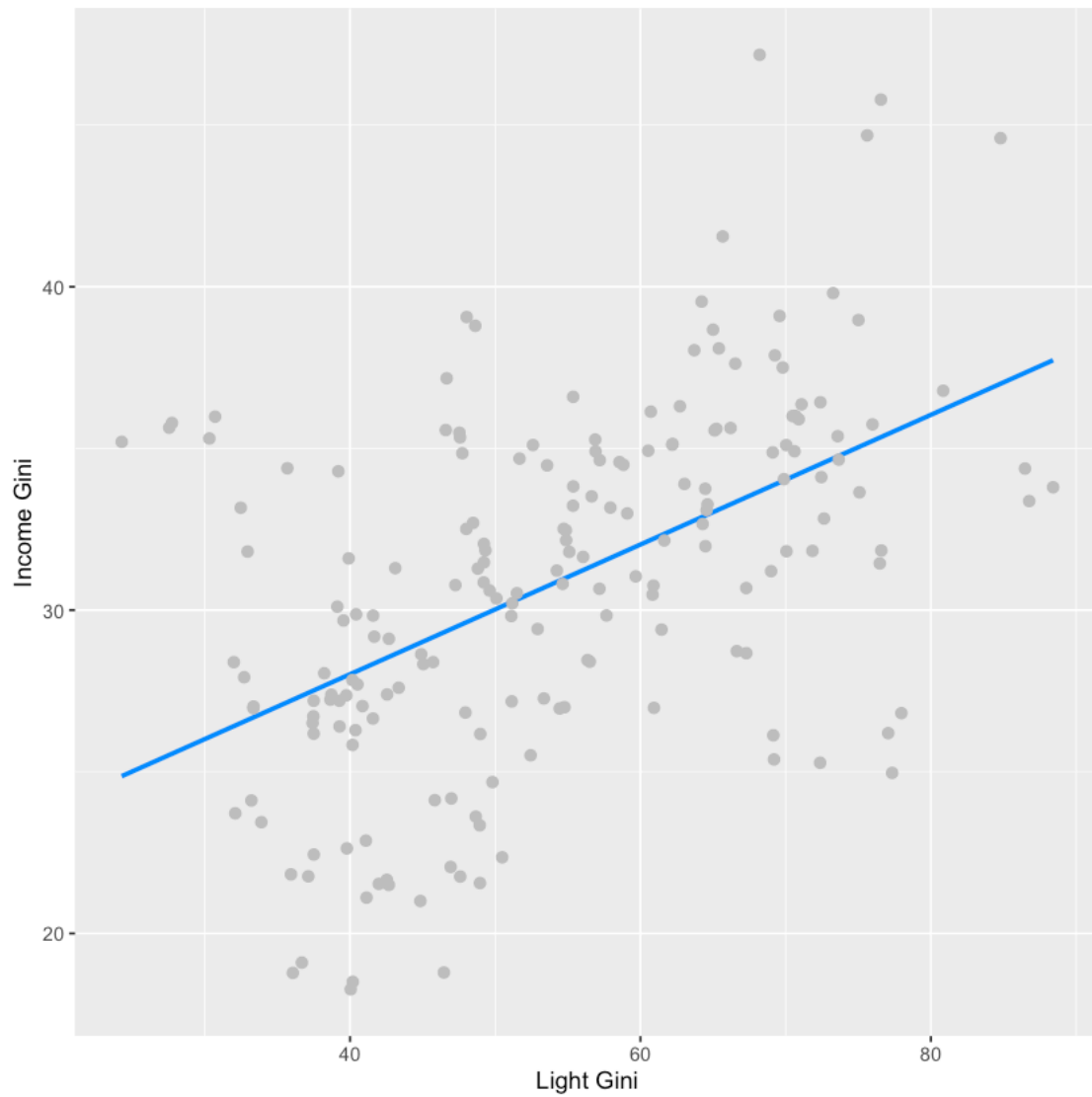
Observations	191	191	191	191
R2				0.690
Adjusted R2				0.685
Log Likelihood	-668.500	-659.514	-577.151	
Akaike Inf. Crit.	1,345.001	1,329.028	1,166.302	
Bayesian Inf. Crit.	1,358.010	1,345.289	1,185.816	
Residual Std. Error				5.155 (df = 187)
F Statistic				138.709*** (df = 3; 187)

Note: *p<0.1; **p<0.05; ***p<0.01

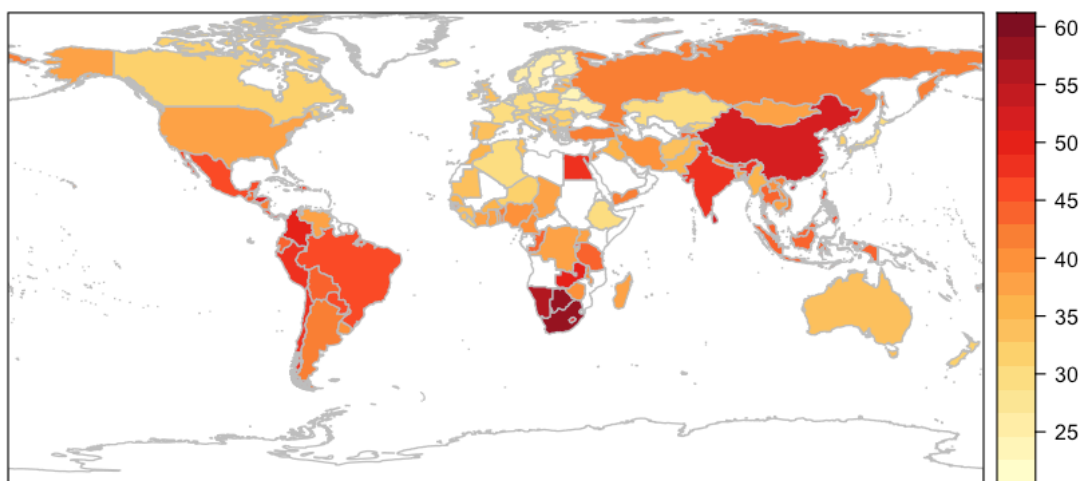
3 Replicar Figura 1

```
[19]: mod_3 <- lmer(Gini ~ Light_Gini +
  log(POP) +
  log(GDP) +
  (1 | Year),
  data = df_nat)
```

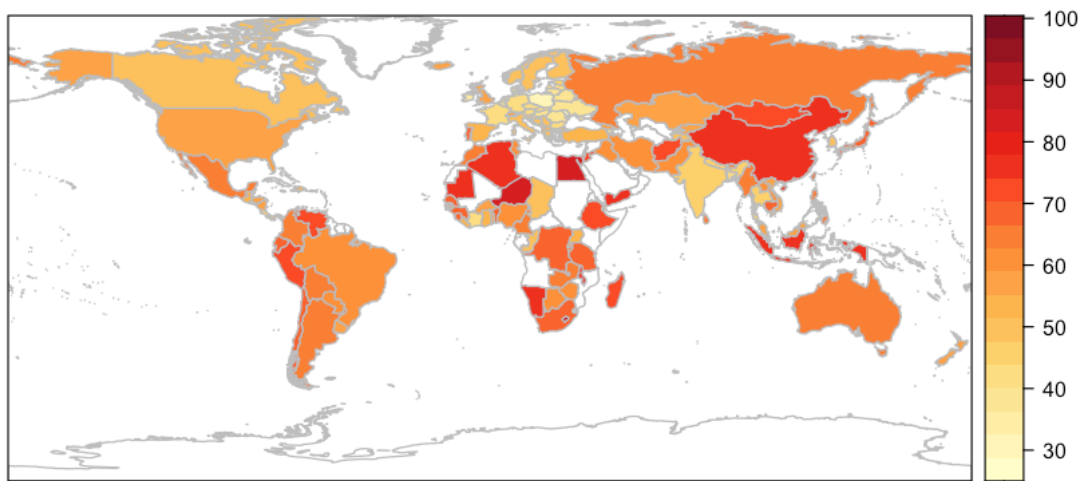
```
[20]: #Fig 1
visreg(mod_3, 'Light_Gini', gg = T, overlay = T, band = F) +
  geom_point(size = 2, colour = 'grey') + labs(x = 'Light Gini', y = 'Income_
↳Gini')
```



```
[21]: spplot(world_2010, c("gini_disp"), col.regions = colorRampPalette(brewer.pal(9, "YlOrRd"))(20),
           cuts = 19, col = "grey")
```



```
[22]: splot(world_2010, c("light_gini_lpp"), col.regions = colorRampPalette(brewer.
↪pal(9, 'YlOrRd'))(20),
      cuts = 19, col = "grey")
```

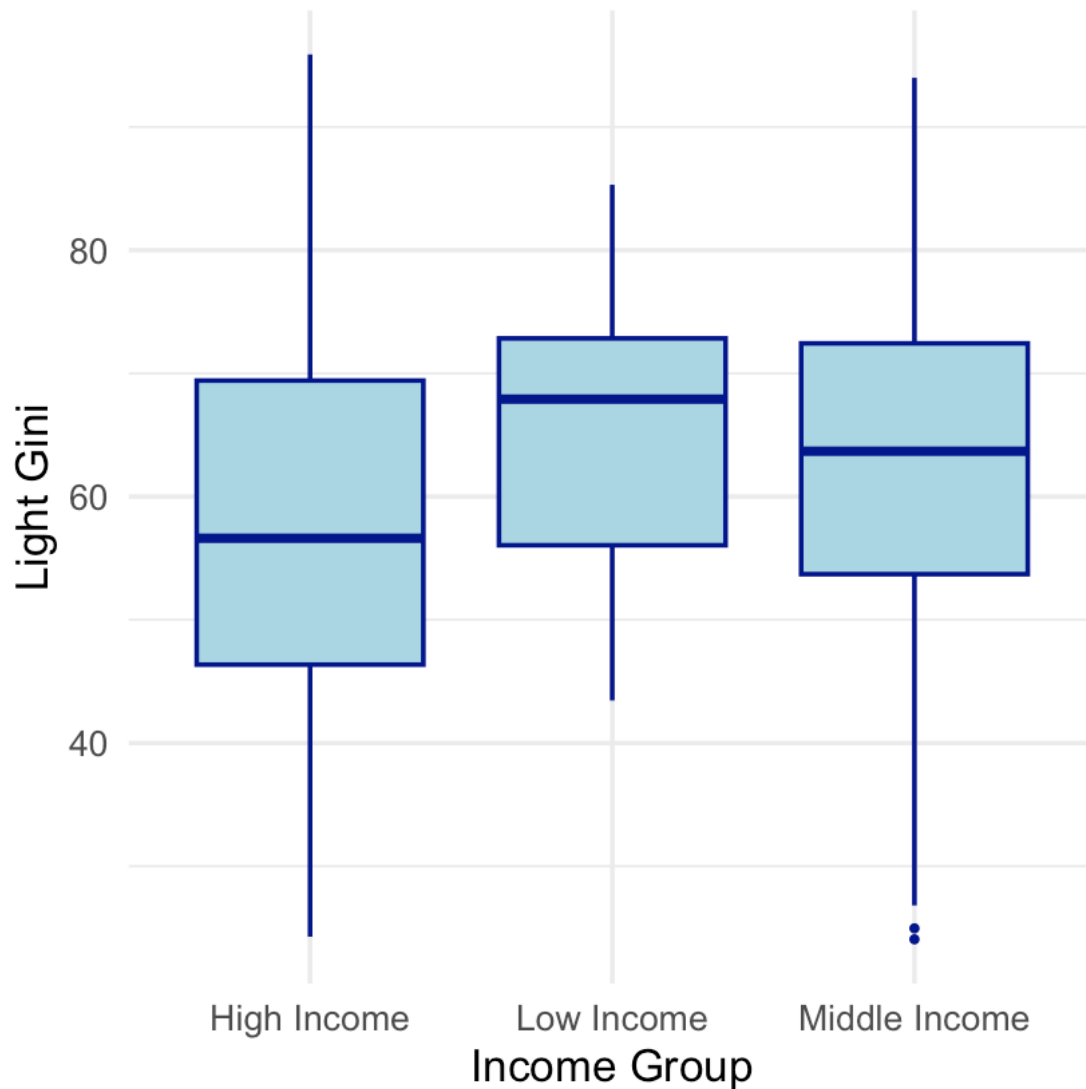


4 Replicar Figura 2

```
[23]: ggplot(data = swiid_df, aes(income_cat, light_gini_lpp)) +
  geom_boxplot(colour = 'darkblue', fill = 'lightblue', size = 1) +
  labs(x = 'Income Group', y = 'Light Gini') +
  theme_minimal(base_size = 20) + xlim(c('High Income', 'Low Income', 'Middle_
↪Income'))
```

Warning message:

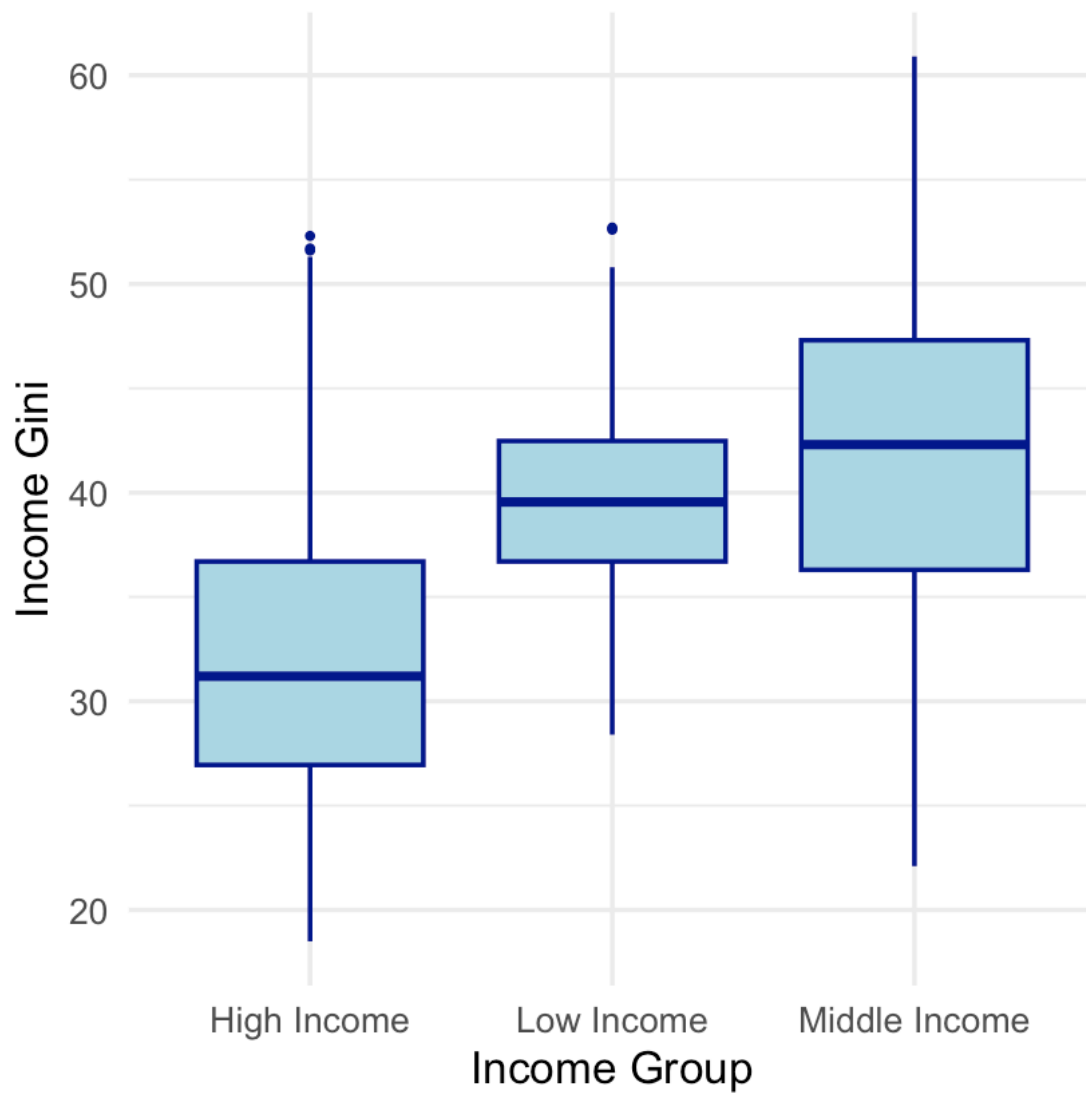
"Removed 13 rows containing missing values (`stat_boxplot()`)."



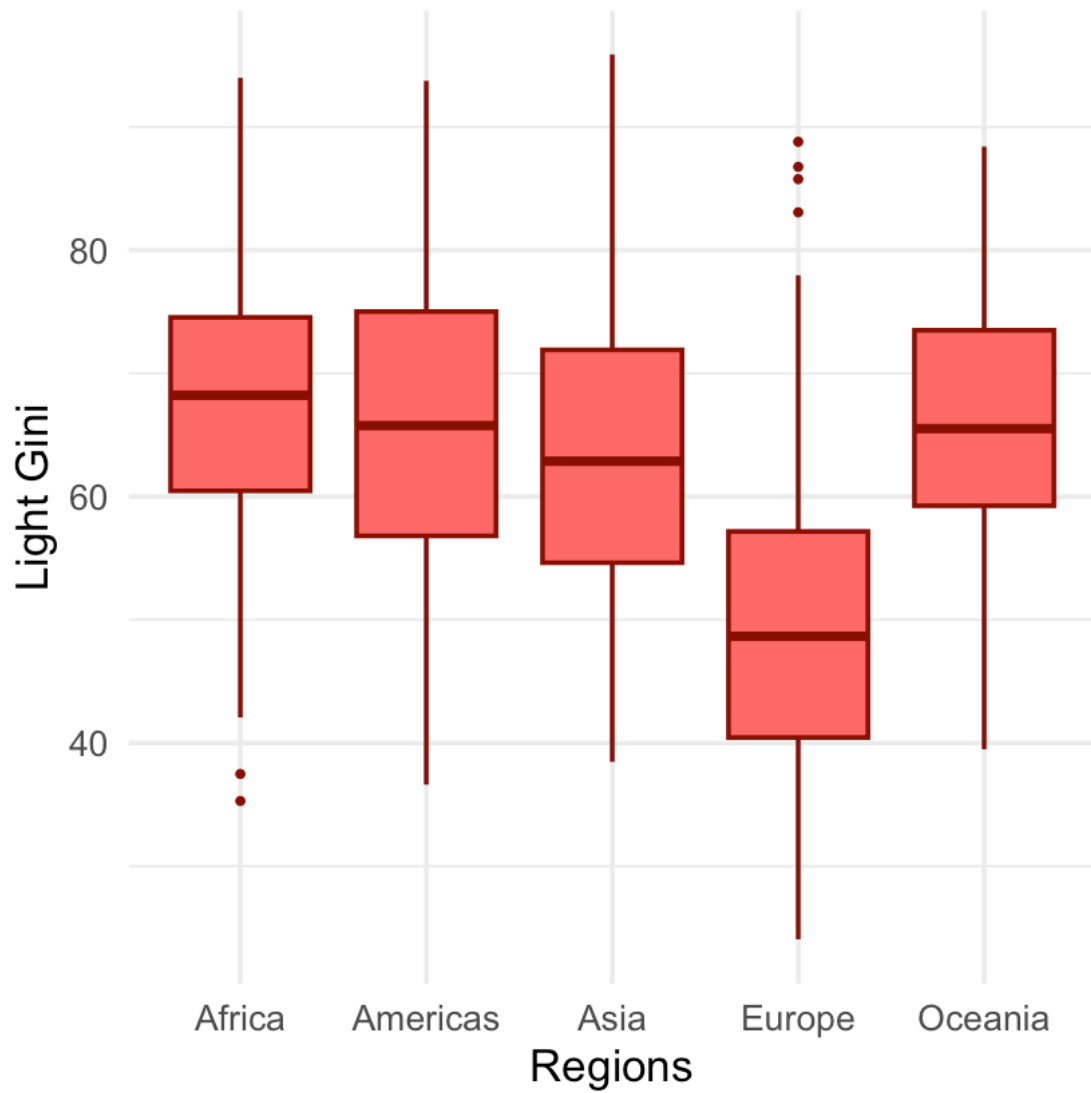
```
[24]: ggplot(data = swiid_df, aes(income_cat, gini_disp)) +
  geom_boxplot(colour = 'darkblue', fill = 'lightblue', size = 1) +
  labs(x = 'Income Group', y = 'Income Gini') +
  theme_minimal(base_size = 20) + xlim(c('High Income', 'Low Income', 'Middle_
↪Income'))
```

Warning message:

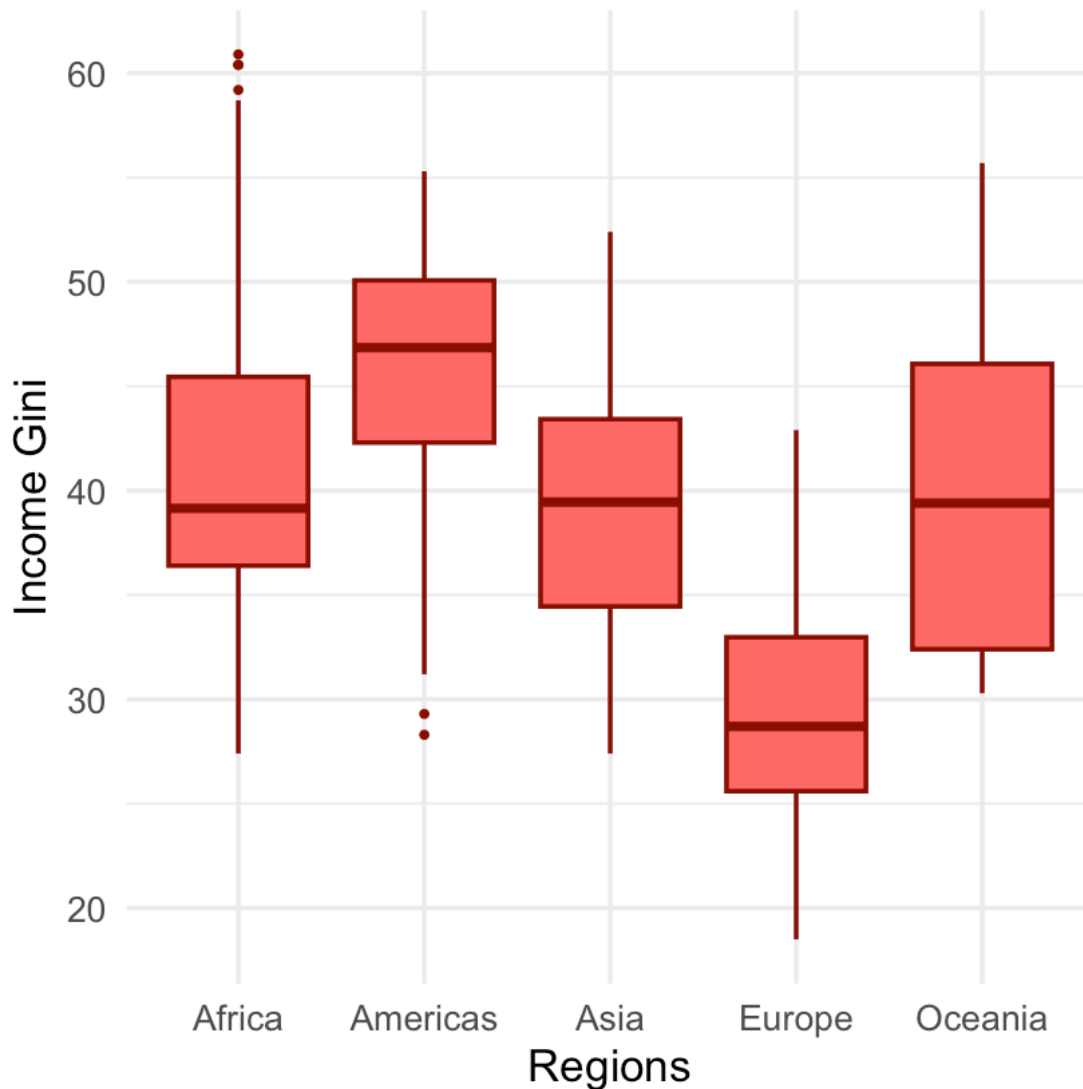
"Removed 13 rows containing missing values (`stat_boxplot()`)."



```
[25]: ggplot(data = swiid_df, aes(region, light_gini_lpp)) +  
  geom_boxplot(colour = 'red4', fill = 'indianred1', size = 1) +  
  labs(x = 'Regions', y = 'Light Gini') +  
  theme_minimal(base_size = 20)
```



```
[26]: ggplot(data = swiid_df, aes(region, gini_disp)) +  
  geom_boxplot(colour = 'red4', fill = 'indianred1', size = 1) +  
  labs(x = 'Regions', y = 'Income Gini') +  
  theme_minimal(base_size = 20)
```



5 Replicar Figura 3

```
[27]: spplot(US_states_10, c("Estimate..Gini.Index"), par.
      ↪ settings=list(fontsize=list(text=20)),
      col.regions = colorRampPalette(brewer.pal(9, 'Reds'))(50), cuts = 49,
      ↪ col = "transparent")

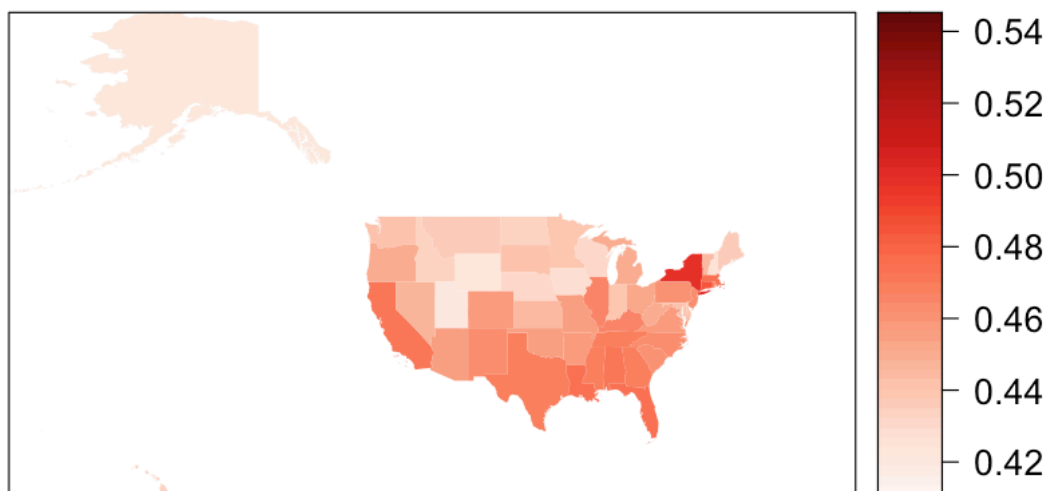
spplot(US_states_10, c("light_gini"), par.settings=list(fontsize=list(text=20)),
      col.regions = colorRampPalette(brewer.pal(9, 'Reds'))(50), cuts = 49,
      ↪ col = "transparent")

# La figura de EEUU por condado no se puede replicar
```

```
# spplot(US_county, c("Estimate..Gini.Index"), par.
  ↳ settings=list(fontsize=list(text=20)),
#       col.regions = colorRampPalette(brewer.pal(9, 'Reds'))(50), cuts = 49,
  ↳ col = "transparent")

cor.test(~ light_gini + Gini, df_US_states[year != 2005])

ggplot(df_US_states[year != 2005], aes(light_gini, Gini)) + geom_point() +
  geom_smooth(method = 'lm', se = F) + labs(x = 'Light Gini', y = 'Income Gini')
```



Pearson's product-moment correlation

```
data: light_gini and Gini
t = 9.2618, df = 253, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4054128 0.5895929
sample estimates:
      cor
0.5031953

`geom_smooth()` using formula = 'y ~ x'
```

