Clase 3 - EPH 2018

Demian Zayat

6/12/2019

En esta clase vamos a trabajar con la Encuesta Anual de Hogares (EAH) de 2018, que realiza periódicamente la Dirección de Estadísticas y Censos del GCBA, disponible en este link https://www.estadisticaciudad.gob.ar/eyc/?p=99567. Primero tendremos que descargar la base, que está disponible públicamente. El archivo Zip contiene dos documentos pdf con la descripción sobre la base y la metodología. El documento ´eah2018_usuarios_documento.pdf´ es clave para conocer la codificación de la base. Luego tiene los archivos ´eah2018_usuarios_ind.txt´ y ´eah2018_usuarios_hog.txt´ con los datos de individuos y hogares encuestados.

Es una muestra estratificada por comunas, que releva información socioeconómica de la población de la Ciudad de Buenos Aires. La base usuaria releva a los individuos, y la base hogares a los hogares. Vamos a usar la de individuos.

Primero vamos a cargar la base y explorarla, a ver su estructura y variables.

```
library(tidyverse)
library(sf)

individuos <- read.csv2("eah2018_usuarios_ind.txt")

str(individuos)
```

```
## 'data.frame':    14497 obs. of  82 variables:
##  $ id          : int  1 2 2 2 3 4 5 5 6 6 ...
##  $ nhogar      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ miembro     : int  1 1 2 3 1 1 1 2 1 2 ...
##  $ comuna      : int  8 9 9 9 2 2 2 2 7 7 ...
##  $ dominio     : int  4 3 3 3 4 4 4 4 4 4 ...
##  $ edad        : int  16 18 37 0 18 18 18 22 18 23 ...
##  $ sexo        : int  1 2 1 1 2 1 2 2 2 2 ...
##  $ parentes_2  : int  1 1 7 3 1 1 1 7 1 7 ...
##  $ p5_2        : int  6 1 1 0 6 6 6 6 6 6 ...
##  $ p6_a        : int  95 95 0 95 95 95 95 95 95 95 ...
##  $ p6_b        : int  95 95 0 1 95 95 95 95 95 95 ...
##  $ estado      : int  1 1 1 3 2 2 3 3 1 1 ...
##  $ categori    : int  3 3 3 0 0 0 0 0 3 3 ...
##  $ t13         : int  0 0 0 0 0 0 2 2 0 0 ...
##  $ t14         : int  0 0 0 0 0 0 2 2 0 0 ...
##  $ t15         : int  0 0 0 0 1 2 0 0 0 0 ...
##  $ t18         : int  0 0 0 0 1 1 0 0 0 0 ...
##  $ t19         : int  0 0 0 0 2018 2018 0 0 0 0 ...
##  $ t28         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ t29         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ t29a        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ t30         : int  1 1 1 0 0 0 0 0 1 1 ...
##  $ sem_hs      : Factor w/ 167 levels "0.00","1.00",..: 60 118 59 1 1 1 1 1 28 119 ...
##  $ t33         : int  2 1 1 0 0 0 0 0 1 2 ...
##  $ t34         : int  0 1 1 0 0 0 0 0 1 0 ...
##  $ t35         : int  2 1 9 0 0 0 0 0 1 1 ...
##  $ t37_cod_2   : int  3100 4810 4803 0 0 0 0 0 4807 4804 ...
```

```
##  $ t37_coda_2  : int  1 3 3 0 0 0 0 0 3 3 ...
##  $ t38         : int  3 2 2 0 0 0 0 0 2 2 ...
##  $ t39         : int  2 5 2 0 0 0 0 0 1 2 ...
##  $ t40         : int  3 2 2 0 0 0 0 0 1 2 ...
##  $ t41_cod_2   : int  80313 33314 30313 0 0 0 0 0 30313 80313 ...
##  $ t47         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ t48         : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ t49         : int  3 1 1 0 0 0 0 0 0 1 ...
##  $ t50a        : int  2 2 2 0 0 0 0 0 0 2 ...
##  $ t50b        : int  2 2 2 0 0 0 0 0 0 2 ...
##  $ t50c        : int  2 2 2 0 0 0 0 0 0 2 ...
##  $ t50d        : int  2 2 2 0 0 0 0 0 0 2 ...
##  $ t50e        : int  2 2 2 0 0 0 0 0 0 2 ...
##  $ t50f        : int  2 2 2 0 0 0 0 0 0 2 ...
##  $ t51_bis     : int  3 3 3 0 0 0 0 0 3 3 ...
##  $ i1          : int  1 1 1 0 0 0 0 0 0 1 ...
##  $ i4          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ i6_3        : int  2 2 2 0 0 0 0 0 0 2 ...
##  $ i10         : int  0 0 0 0 0 0 0 0 2 0 ...
##  $ i11         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ codioph     : int  1 1 9 0 0 0 0 0 1 1 ...
##  $ ioph_2      : int  2000 500 12000 0 0 0 0 0 500 1500 ...
##  $ codioph_neto: int  1 1 9 0 0 0 0 0 1 1 ...
##  $ ioph_neto_2 : int  2000 500 12000 0 0 0 0 0 500 1500 ...
##  $ codios      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ios_2       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ codioa      : int  2 2 2 0 2 2 0 0 2 2 ...
##  $ ioa_2       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ codlab      : int  1 1 9 0 2 2 2 2 1 1 ...
##  $ inglab_2    : int  2000 500 12000 0 0 0 0 0 500 1500 ...
##  $ codnolab    : int  2 2 1 0 1 1 1 1 9 9 ...
##  $ ingnolab_2  : int  0 0 5000 0 6000 16500 8000 8000 300 13000 ...
##  $ coding      : int  1 1 9 0 1 1 1 1 9 9 ...
##  $ ingtot_2    : int  2000 500 17000 0 6000 16500 8000 8000 800 14500 ...
##  $ codi_tot    : int  1 9 9 9 1 1 1 1 9 9 ...
##  $ itfb_2      : int  2000 17500 17500 17500 6000 16500 16000 16000 15300 15300 ...
##  $ ipcfb_2     : int  2000 5833 5833 5833 6000 16500 8000 8000 7650 7650 ...
##  $ e2          : int  2 1 2 3 1 1 1 1 2 2 ...
##  $ e4          : int  0 1 0 0 1 3 1 1 0 0 ...
##  $ e6          : int  0 10 0 0 13 13 13 13 0 0 ...
##  $ e12         : int  7 0 3 0 0 0 0 0 7 7 ...
##  $ nivel       : int  4 4 2 8 6 6 6 6 4 4 ...
##  $ aesc        : int  7 8 3 0 12 12 12 16 9 11 ...
##  $ m1          : int  5 1 1 1 2 4 2 2 5 5 ...
##  $ m1_2        : int  5 1 1 1 2 4 2 2 5 5 ...
##  $ m2_anio     : int  2009 0 0 0 0 2018 0 0 2016 2013 ...
##  $ m3_anio     : int  2009 1 1 1 2018 2018 2018 2014 2016 2013 ...
##  $ tipcob2_2   : int  1 1 1 1 2 1 2 2 1 1 ...
##  $ s2          : int  1 1 1 1 1 3 1 1 1 2 ...
##  $ sn4         : int  3 3 2 3 3 3 3 3 3 3 ...
##  $ sn5         : int  1 1 3 2 7 6 6 5 2 2 ...
##  $ sn16        : int  3 4 4 3 2 2 1 1 3 2 ...
##  $ s28         : int  0 1 0 0 2 0 2 2 2 1 ...
##  $ s29         : int  0 1 0 0 0 0 0 0 0 1 ...
```

```
##  $ fexp        : int  55 40 40 40 214 180 196 196 239 239 ...
```

```
names(individuos)
```

```
##  [1] "id"           "nhogar"       "miembro"      "comuna"
##  [5] "dominio"      "edad"         "sexo"         "parentes_2"
##  [9] "p5_2"         "p6_a"         "p6_b"         "estado"
## [13] "categori"     "t13"          "t14"          "t15"
## [17] "t18"          "t19"          "t28"          "t29"
## [21] "t29a"         "t30"          "sem_hs"       "t33"
## [25] "t34"          "t35"          "t37_cod_2"    "t37_coda_2"
## [29] "t38"          "t39"          "t40"          "t41_cod_2"
## [33] "t47"          "t48"          "t49"          "t50a"
## [37] "t50b"         "t50c"         "t50d"         "t50e"
## [41] "t50f"         "t51_bis"      "i1"           "i4"
## [45] "i6_3"         "i10"          "i11"          "codioph"
## [49] "ioph_2"       "codioph_neto" "ioph_neto_2"  "codios"
## [53] "ios_2"        "codioa"       "ioa_2"        "codlab"
## [57] "inglab_2"     "codnolab"     "ingnolab_2"   "coding"
## [61] "ingtot_2"     "codi_tot"     "itfb_2"       "ipcfb_2"
## [65] "e2"           "e4"           "e6"           "e12"
## [69] "nivel"        "aesc"         "m1"           "m1_2"
## [73] "m2_anio"      "m3_anio"      "tipcob2_2"    "s2"
## [77] "sn4"          "sn5"          "sn16"         "s28"
## [81] "s29"          "fexp"
```

```
dim(individuos)
```

```
## [1] 14497    82
```

Vemos que es una base de datos de 14497 observaciones en 82 variables. Es demasiado para manipularlas facilmente. Así que mejor analizamos el glosario con la codificación que viene en PDF, en el archivo "eah2018_usuarios_documento.pdf". En este proyecto nos vamos a centrar en analizar solo algunas variables de la base usuarios individuos, ustedes pueden despues armar análisis alternativos:

- Comuna
- Dominio: Vivienda en villa o no
- Edad
- Sexo
- Estado (condición de actividad)
- ingtot_2 (ingresos totales)
- ipc_fb2 (ingreso per capita familiar)
- nivel (nivel educativo alcanzado)
- aesc (años de escolaridad)
- m1_2 (lugar de nacimiento)
- tipcob2_2 (cobertura de salud)
- fexp (factor de expansión)

Para ello, debemos seleccionar sólo esas variables y armar una base más chica

```
individuos_chica <- individuos %>%
  select(comuna, dominio, edad, sexo, estado, ingtot_2, ipcfb_2, nivel, aesc, m1_2, tipcob2_2, fexp)
```

Ahora tenemos 14497 observaciones, pero de 12 variables nomás.

Población por sexo

Según el documento metodológico, la muestra de 14.497 individuos representa a los 3.067.990 individuos que habitan en la Ciudad de Buenos Aires. El factor de expansión de la muestra estratificada lo realiza la variable `fexp`, de modo que cada observación representa tanta población como indica el valor `fexp`. Si sumamos eso, nos debería dar toda la ciudad. Probemos:

```
sum(individuos_chica$fexp)
```

```
## [1] 3067990
```

Si queremos ver cuántos varones y cuántas mujeres hay en la Ciudad podemos hacerlo con el verbo `summarize`, asociado al `group_by` con el siguiente código. Primero podríamos recodificar la base para hacerla más facil de comprender.

```
individuos_chica <- individuos_chica %>%
  mutate(sexo = recode(sexo,  `1` ="Varones",
                               `2` = "Mujeres"))

xsexo <- individuos_chica %>%
  group_by(sexo) %>%
  summarize(cant = sum(fexp)) %>%
  mutate(porc = round(cant/sum(cant)*100,2))
xsexo
```

```
## # A tibble: 2 x 3
##   sexo       cant  porc
##   <chr>     <int> <dbl>
## 1 Mujeres 1630072  53.1
## 2 Varones 1437918  46.9
```

Ahora podemos medir la tasa de actividad, que mide la variable `estado`. Simplemente agrupamos por ´sexo´ y por ´estado´. Antes podemos recodificar la base con los estados.

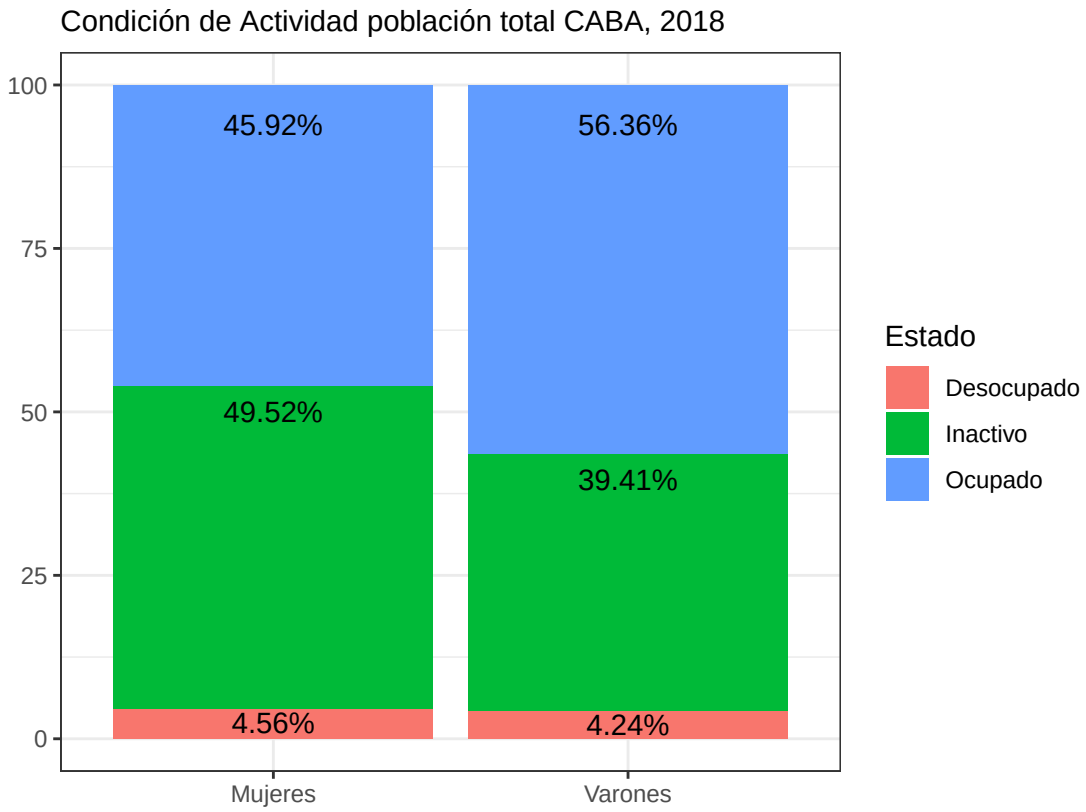
```
individuos_chica <- individuos_chica %>%
  mutate(estado = recode(estado,  `1` ="Ocupado",
                         `2` = "Desocupado",
                         `3` = "Inactivo"))
xactividad <- individuos_chica %>%
  group_by(sexo, estado) %>%
  summarize (cant = sum(fexp)) %>%
  mutate(porc = round(cant/sum(cant)*100,2))

xactividad
```

```
## # A tibble: 6 x 4
## # Groups:   sexo [2]
##   sexo    estado       cant  porc
##   <chr>   <chr>       <int> <dbl>
## 1 Mujeres Desocupado  74264  4.56
## 2 Mujeres Inactivo   807284 49.5
## 3 Mujeres Ocupado    748524 45.9
## 4 Varones Desocupado  60917  4.24
## 5 Varones Inactivo   566627 39.4
## 6 Varones Ocupado    810374 56.4
```

Y lo graficamos

```
ggplot(data = xactividad, aes(x = sexo, y = porc, group = sexo))+
  geom_col(aes(fill = estado))+
  geom_text(aes(label = paste0(porc,"%"), y = porc-2), position = "stack")+
  labs(subtitle = "Condición de Actividad población total CABA, 2018",
       y = "", x= "", fill = "Estado")+
  theme_bw()
```



Ingresos

Los ingresos mostrarán alguna diferencia por sexo? Debemos cambiar la variable de agrupamiento de estado por ingresos

```
xingresos <- individuos_chica %>%
  group_by(sexo) %>%
  summarize(prom = weighted.mean(ingtot_2, fexp))
xingresos <- xingresos %>%
  mutate(porc = round(prom/xingresos$prom[2]*100,2))
```

Y podemos analizar los ingresos por comuna

```
xingresos_comuna <- individuos_chica %>%
  group_by(comuna) %>%
  summarize(prom = weighted.mean(ingtot_2, fexp))
xingresos
```

```
## # A tibble: 2 x 3
##   sexo      prom  porc
##   <chr>    <dbl> <dbl>
```

```
## 1 Mujeres 16218.  76.2
## 2 Varones 21298. 100
```

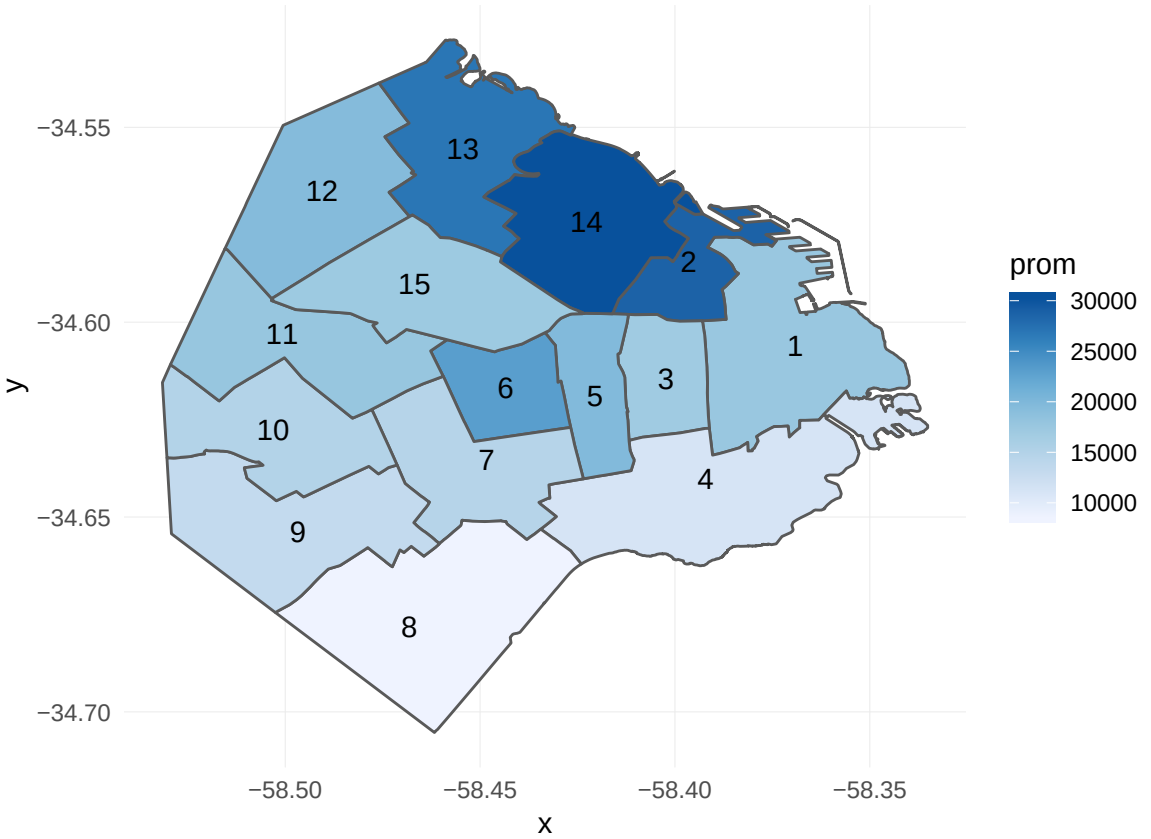
Y hacemos un mapa!

```
comunas <- st_read("/Users/demian/Documents/GitHub/derecho_y_datos/Clase1/CABA_comunas.geojson")
```

```
## Reading layer `OGRGeoJSON' from data source `/Users/demian/Documents/GitHub/derecho_y_datos/Clase1/C
## Simple feature collection with 15 features and 4 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: -58.53152 ymin: -34.70529 xmax: -58.33514 ymax: -34.52754
## epsg (SRID):    4326
## proj4string:    +proj=longlat +datum=WGS84 +no_defs
```

```
#convierto las comunas a factores
xingresos_comuna$comuna <- factor(xingresos_comuna$comuna)
#agrego la geometry al xingresos_comuna
xingresos_comuna_geo <- xingresos_comuna %>%
  left_join(comunas, by = c("comuna" = "comunas"))

#y grafico
ggplot()+
  geom_sf(data = xingresos_comuna_geo, aes(fill = prom))+
  geom_sf_text(data = xingresos_comuna_geo, aes(label = comuna))+
  scale_fill_distiller(direction = 1)+
  theme_minimal()
```

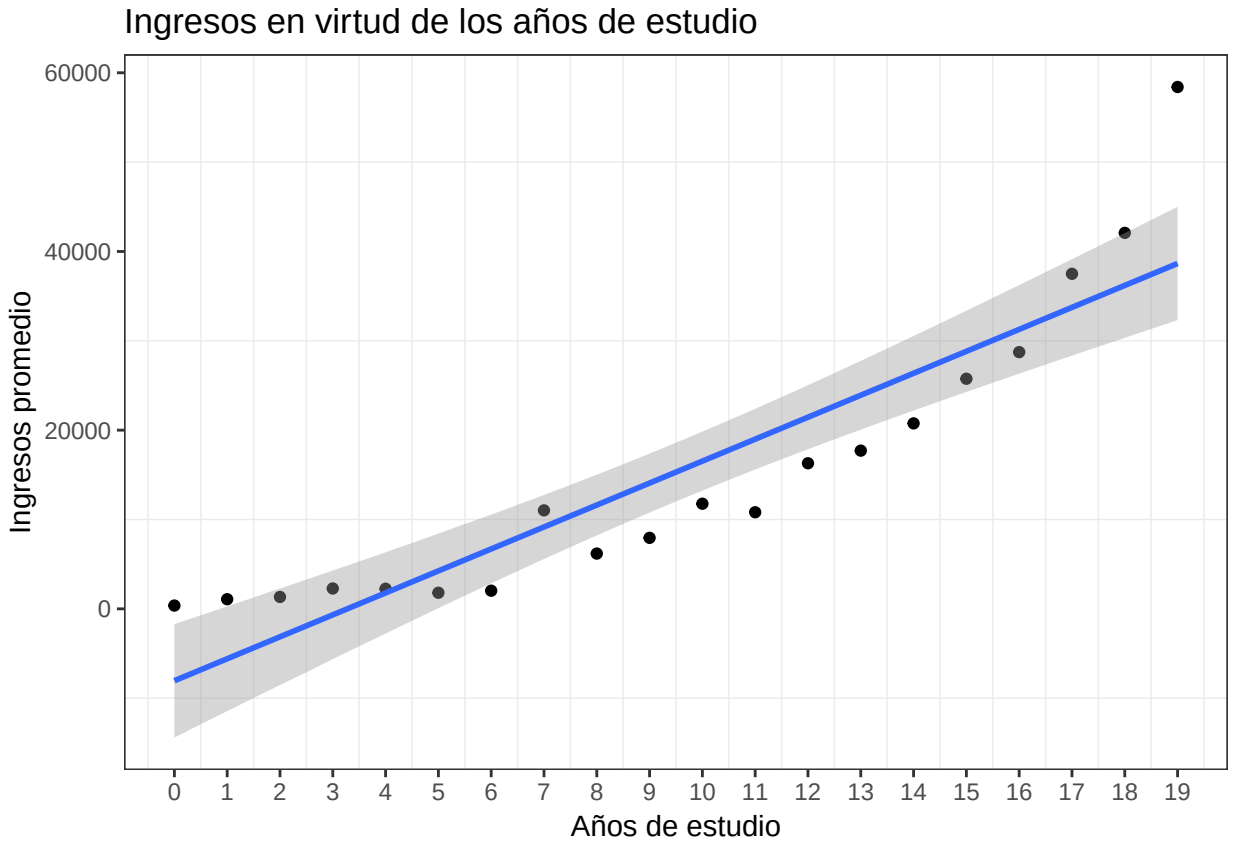


Existe relación entre años de estudio e ingresos?

Y por último, podemos analizar esta pregunta tambien en base a los datos de la base.

```
anos <- individuos_chica %>%
  filter(aesc != 99) %>%
  group_by(aesc) %>%
  summarize (prom = weighted.mean(ingtot_2, fexp))

ggplot(data = anos, group = aesc)+
  geom_point(aes(x = aesc, y =  prom))+
  geom_smooth(aes(x = aesc, y = prom), method = "lm")+
  labs(title = "Ingresos en virtud de los años de estudio",
       x = "Años de estudio", y = "Ingresos promedio")+
  scale_x_continuous(breaks = seq(0,20,1))+
  theme_bw()+
  theme(panel.grid.major = element_blank())
```




```
lm(ingtot_2~aesc, data = individuos_chica)
```

```
##
## Call:
## lm(formula = ingtot_2 ~ aesc, data = individuos_chica)
##
## Coefficients:
## (Intercept)         aesc
##      8568.0        784.8
```