

Datos de Panel

Aplicación al Mercado Laboral con la EPH

Miriam Malament

Economía Laboral, UCEMA

Importar bases

Nos interesa tomar cuatro bases consecutivas. En este caso, vamos a estar usando el segundo, tercer y cuarto trimestre del 2021 y el primer trimestre del 2022.

```
library(eph)
library(tidyverse)
individual.221 ← get_microdata(2021,2)
individual.321 ← get_microdata(2021,3)
individual.421 ← get_microdata(2021,4)
individual.122 ← get_microdata(2022,1)
```

El objetivo de este análisis es dar un ejemplo conceptual para usar datos de panel en la práctica. Intentaremos contestar qué tan probable es pasar de una categoría ocupacional a otra en un trimestre.

Es importante tener en cuenta que no veremos mucha movilidad si tan solo tomamos cuatro trimestres. Este análisis puede extenderse a varios años.

Código para el Panel

.1. Creamos un Vector que contenga unicamente las variables de interés, para recortar luego la base con la funcion **select**.

```
#Paso 1
var.ind ← c('CODUSU', 'NRO_HOGAR', 'COMPONENTE', 'ANO4', 'TRIMESTRE', 'ESTADO', 'CAT_OCUP',
            'PONDERA', 'CH04', 'CH06', 'P21', 'PP3E_TOT')
```

.2. Unimos todas las bases con la función **bind_rows**, seleccionando solo las variables del vector.

```
#Paso 2
Bases_Continua ← bind_rows(
  individual.221 %>% select(var.ind),
  individual.321 %>% select(var.ind),
  individual.421 %>% select(var.ind),
  individual.122 %>% select(var.ind))
```

Código para el Panel

.3. Filtramos de la base los casos de no respuesta y acotaremos en este ejercicio el análisis a la población entre 18 y 65 años.

.4. Creamos las categorías de análisis que deseamos observar en distintos períodos.

```
#Pasos 3 y 4
Bases_Continua ← Bases_Continua %>%
  filter(CH06 %in% c(18:65), ESTADO ≠ 0) %>%
  mutate(Categoria = case_when(ESTADO %in% c(3,4) ~ "Inactivos",
                                ESTADO = 2 ~ "Desocupados",
                                ESTADO = 1 & CAT_OCUP = 1 ~ "Patrones",
                                ESTADO = 1 & CAT_OCUP = 2 ~ "Cuenta Propistas",
                                ESTADO = 1 & CAT_OCUP = 3 ~ "Asalariados",
                                ESTADO = 1 & CAT_OCUP = 4 ~ "Trabajador familiar s,
                                TRUE ~ "Otros"))
```

Código para el Panel

.5. Armamos un *identificador ordinal* para los registros de cada trimestre.

```
#Paso 5
Bases_Continua ← Bases_Continua %>%
  mutate(Trimestre = paste(ANO4, TRIMESTRE, sep="_")) %>%
  arrange(Trimestre) %>%
  mutate(Id_Trimestre = match(Trimestre, unique(Trimestre)))
```

.6. Replicamos el dataframe construido y le cambiamos los nombres a todas las variables, a excepción de las que usaremos para identificar a un mismo individuo (*CODUSU*, *NRO_HOGAR*, *COMPONENTE*).

```
#Paso 6
Bases_Continua_Replica ← Bases_Continua
names(Bases_Continua_Replica)

names(Bases_Continua_Replica)[4:(length(Bases_Continua_Replica)-1)] ←
  paste0(names(Bases_Continua_Replica)[4:(length(Bases_Continua_Replica)-1)], "_t1")
names(Bases_Continua_Replica)
```

Código para el Panel

.7. En la base replicada, modificamos el *identificador ordinal* en función de la amplitud que deseamos en las observaciones de panel. En nuestro caso como uniremos registros con distancia de 1 trimestre, le restamos 1 a cada valor

```
#Paso 7
```

```
Bases_Continua_Replica$Id_Trimestre ← Bases_Continua_Replica$Id_Trimestre - 1
```

.8. Unimos ambas bases con la función **inner_join** que solo nos dejará registros que en ambas bases contengan los mismos *CODUSU*, *NRO_HOGAR*, *COMPONENTE* e *identificador ordinal*.

.9. Creamos la columna para las consistencias, y luego filtramos la base para eliminar los registros inconsistentes

```
#Pasos 8 y 9
```

```
Panel_Continua ← inner_join(Bases_Continua, Bases_Continua_Replica)
```

```
Panel_Continua ← Panel_Continua %>%
```

```
  mutate(Consistencia = case_when(abs(CH06_t1-CH06) > 2 |  
                                   CH04 ≠ CH04_t1 ~ "inconsistente",  
                                   TRUE ~ "consistente")) %>%
```

```
  filter(Consistencia = "consistente")
```

Matriz de Transición

Calculamos la probabilidad de transición.

```
#Matrices de transición
## Calculo de probabilidades de transición.

Categorias_transiciones <- Panel_Continua %>%
  #filter(Categoria != Categoria_t1) %>%
  group_by(Categoria, Categoria_t1) %>%
  summarise(frec_muestral = n(),
            frecuencia = sum((PONDERA+PONDERA_t1)/2)) %>%
  ungroup() %>%
  group_by(Categoria) %>%
  mutate(Prob_salida = frecuencia/sum(frecuencia))
```

Matriz de Transición

Categorias_transiciones

```
# A tibble: 36 × 5
```

```
# Groups:   Categoria [6]
```

	Categoria	Categoria_t1	frec_muestral	frecuencia	Prob_salida
	<chr>	<chr>	<int>	<dbl>	<dbl>
1	Asalariados	Asalariados	14738	7704478.	0.886
2	Asalariados	Cuenta Propistas	734	363822	0.0418
3	Asalariados	Desocupados	341	163288	0.0188
4	Asalariados	Inactivos	843	407878	0.0469
5	Asalariados	Patrones	99	52085	0.00599
6	Asalariados	Trabajador familiar s/r	11	7730.	0.000889
7	Cuenta Propistas	Asalariados	818	434797	0.159
8	Cuenta Propistas	Cuenta Propistas	3377	1786675	0.654
9	Cuenta Propistas	Desocupados	179	96818	0.0354
10	Cuenta Propistas	Inactivos	567	292765	0.107

```
# ... with 26 more rows
```

```
# i Use `print(n = ...)` to see more rows
```


Matriz de Transición

```
library(ggthemes)
plot ← ggplot(Categorias_transiciones, aes(x = Categoria_t1,
                                             y = Categoria, fill = Prob_salida,
                                             label =round(Prob_salida*100,2))) +
  labs(title = "Probabilidades de Transicion de hacia las distintas Categorías")+
  geom_tile()+
  geom_text()+
  scale_fill_gradient(low = "grey100", high = "grey30")+
  theme_tufte()
```

Matriz de Transición

