
PyEPH

Versión v1.1.3

Maria Carolina Trogliero, Mariano Valdez Anopa, Maria Frances G

12 de abril de 2023

1. Cómo citar la librería	3
2. Tenga en cuenta	5
3. Agradecimientos	7
4. Instalación	9
4.1. Prerequisitos	9
4.2. Instalando PyEPH	9
5. Uso básico	11
6. Obtención de bases de datos	13
6.1. ¿Qué se obtiene?	13
6.2. ¿En qué formato se obtienen?	13
6.3. ¿Desde dónde se obtienen?	13
6.4. ¿Dónde se almacenan?	14
6.5. Uso	14
7. Cálculos	17
7.1. Cálculo de la Pobreza e Indigencia	17
7.2. Cálculos de Mercado Laboral	19

La librería Pyeph tiene como objetivo facilitar el procesamiento en Python de las [Encuesta Permanente de Hogares \(eph\)](#) publicadas por INDEC de forma periódica. Está pensada como un espacio donde se nuclean y centralizan los cálculos vinculados a las mismas para posteriormente ser utilizadas en investigaciones, artículos, publicaciones, etc. Es una librería que hace principal hincapié en la transparencia metodológica utilizando licencias de código abierto y que promueve la colaboración de las comunidades de científicos de datos, sociales, investigadorxs, desarrolladorxs, periodistas y demás curiosxs.

Permite la descarga de archivos de EPH's y otros como la [canasta basica y adulto equivalente](#) , como asi también algunos calculos rápidos relacionados con las mismas

Repo: <https://github.com/institutohumai/pyeph>

CAPÍTULO 1

Cómo citar la librería

Carolina Trogliero, Mariano Valdez y Maria Frances Gaska (2022). PyEPH: Librería para la
↳obtención y el procesamiento de la Encuesta Permanente de Hogares (EPH-INDEC). PyEPH.
↳version <https://doi.org/10.5281/zenodo.6727908>

CAPÍTULO 2

Tenga en cuenta

Esta librería se encuentra en estado permanente de desarrollo.

Cualquier colaboración es bienvenida

CAPÍTULO 3

Agradecimientos

Dejamos aquí un especial agradecimiento al equipo de desarrollo de la librería [EPH en R](#). Todo el amor para ellos y a [Rami Argañaraz](#) por armarnos el loguito

Pueden probar nuestra notebook de ejemplo en Google Colab

Recordá abrir en una nueva pestaña

4.1 Prerequisitos

- Python 3
- pip

4.2 Instalando PyEPH

- Abra una terminal del sistema y escriba

```
$ pip install pyeph
```


Los siguientes son algunos ejemplos de uso. Para ver todos los cálculos podés ir para la documentación

En inglés

```
import pyeph

# Obtención
eph = pyeph.get(data="eph", year=2021, period=2, base_type='individual') # EPH individual
basket = pyeph.get(data="canastas") # canasta basica total y alimentaria
adequi = pyeph.get(data="adulto-equivalente") # adulto equivalente

# Cálculos de ejemplo de pobreza
poverty = pyeph.Poverty(eph, basket)
population_poverty = poverty.population(group_by='CH04') # Población pobre por sexo
labeled_poverty = pyeph.map_labels(population_poverty) # Etiquetado de las variables

# Cálculos de Mercado Laboral
labor_market = pyeph.LaborMarket(eph)
unemployment = labor_market.unemployment(group_by="REGION", div_by="PT") # Desempleo
↳ agrupado por region y dividiendo por Población Total
labeled_unemployment = pyeph.map_labels(unemployment) # Etiquetado de las variables
```

En español

```
import pyeph

# Obtención
eph = pyeph.obtener(data="eph", ano=2021, periodo=2, tipo_base='individual') # EPH
↳ individual
canastas = pyeph.obtener(data="canastas") # canasta basica total y alimentaria
adequi = pyeph.obtener(data="adulto-equivalente") # adulto equivalente
```

(continué en la próxima página)

(proviene de la página anterior)

```
# Cálculos de ejemplo de pobreza
pobreza = pyeph.Pobreza(eph, canastas)
poblacion_pobre = pobreza.poblacion(agrupar_por='CH04') # Población pobre por sexo
poblacion_pobre_etiquetado = pyeph.etiquetar(poblacion_pobre) # Etiquetado de las
↳ variables

# Cálculos de Mercado Laboral
mercado_laboral = pyeph.MercadoLaboral(eph)
desempleo = mercado_laboral.desempleo(agrupar_por="REGION", div_por="PT") # Desempleo
↳ agrupado por region y dividiendo por Población Total
desempleo_etiquetado = pyeph.etiquetar(desempleo) # Etiquetado de las variables
```

Obtención de bases de datos

La función `pyeph.obtener()` o `pyeph.get()` - genera la obtención de un `pandas.DataFrame` de las bases que el usuario solicite.

6.1 ¿Qué se obtiene?

- Microdata
- Canasta básica
- Adulto equivalente
- En construcción...

6.2 ¿En qué formato se obtienen?

Los archivos originales se encuentran en formato “.csv” comprimidos en formato “.zip” Lo que se almacena efectivamente en disco es el archivo “.zip”

6.3 ¿Desde dónde se obtienen?

Los archivos se encuentran almacenados en github. Esto debido a la necesidad de estandarizar los archivos para su posterior procesamiento en los cálculos. Los archivos servidos por INDEC se encuentran descentralizados y con diferentes nomenclaturas. Y, además, de esta manera ahorramos el posible mal funcionamiento del módulo de descarga de la librería puesto que no siempre está disponible el sitio de INDEC para su consulta. En cualquier caso, de verse necesario realizar la obtención de los datos desde INDEC por asuntos de fiabilidad, es un llamamiento a la comunidad para desarrollar esa sección. En `getter.py` se deberá modificar el método `from_indec()`

6.4 ¿Dónde se almacenan?

Los archivos descargados se pueden encontrar dentro de la carpeta «.db», localizada dentro de «ruta-local-de-python/site-packages/pyeph/»

6.5 Uso

Dependiendo el tipo de base que se quiera obtener, se realizarán diferentes acciones.

6.5.1 Microdatos

Al peticionar los microdatos pyeph revisa si existe el archivo de la eph solicitada en local. En caso de que exista, retorna el DataFrame del archivo .csv; en caso de no existir primero realiza su descarga.

```
pyeph.obtener(data="eph", ano=2017, periodo=2, frecuencia="trimestre", tipo_base=
→ "individual")

o

pyeph.get(data="eph", year=2017, period=2, freq="trimestre", base_type="individual")
```

Parámetros

Parámetros	Tipo de dato	Descripción
data=eph	str	«eph» para solicitar los microdatos de las EPH's
ano (year)	int	Año de la EPH solicitada
periodo (period)	int	Periodo de la EPH solicitada
frecuencia (freq)	str	Frecuencia de periodización (default: «trimestre»)
tipo_base (base_type)	str	Tipo de base (Hogar o Individual) (default: «individual»)

6.5.2 Canasta Básica Alimentaria y Canasta Básica Total

Al peticionar la canasta pyeph intenta obtener las CBA y CBT más actualizadas. Retornará la ultima que encuentre

```
pyeph.obtener(data="canastas")

o

pyeph.get(data="basket")
```

Parámetros

Parámetros	Tipo de dato	Descripción
data=canastas (basket)	str	CBA y CBT

7.1 Cálculo de la Pobreza e Indigencia

La clase `pyeph.Pobreza()` o `pyeph.Poverty()` - instancia el calculo de la pobreza e indigencia utilizando la metodología de línea para las bases trimestrales. Tiene dos métodos principales (`hogares()/household()` y `poblacion()/population()`) que devuelven un dataframe con información referida a la tasa de pobreza de personas y la tasa de pobreza de hogares.

Aclaración: No existe información publicada fuera de los informes de prensa en formato pdf sobre los valores de las canastas básicas y alimentarias. No obstante, hemos desarrollado dos funcionaes que, de encontrarse disponibles dichos datos, podrían calcular de forma automática los valores de pobreza e indigencia.

7.1.1 Uso

La clase `pyeph.Pobreza()` o `pyeph.Poverty()` recibe los siguientes parámetros:

```
pyeph.Pobreza(eph, canasta)
```

o

```
pyeph.Poverty(eph, basket)
```

7.1.2 Parámetros

Parámetros	Tipo de dato	Descripción
eph	pandas.DataFrame	Base de EPH individual publicada por INDEC.
basket (canasta)	pandas.DataFrame	Base de datos de Canasta Básica Total y Canasta Básica Alimentaria con el formato especificado aquí .

7.1.3 Métodos

La clase `pyeph.Pobreza()` o `pyeph.Poverty()` instancia el calculo de pobreza e indigencia. Para obtener el dataframe con los resultados podrá consultar los métodos `hogares()/household()` o `poblacion()/population()`, según corresponda.

- Pobreza e Indigencia de Personas

Este método retorna como resultado un `pandas.DataFrame` donde se especifica el porcentaje y la cantidad de personas en situación de pobreza e indigencia. Esta información puede calcularse agrupada por otras categorías de interés (ejemplo: sexo).

```
pyeph.Pobreza(eph, canasta).poblacion(agrupar_por)

o

pyeph.Poverty(eph, basket).population(group_by)
```

- Pobreza e Indigencia de Hogares

Este método retorna como resultado un `pandas.DataFrame` donde se especifica el porcentaje y la cantidad de hogares en situación de pobreza e indigencia. En este caso, no es posible obtener la información agrupada por otra categoría.

```
pyeph.Pobreza(eph, canasta).hogares()

o

pyeph.Poverty(eph, basket).household()
```

7.1.4 Ejemplos

Primero se instancia la clase `pyeph.Pobreza()` o `pyeph.Poverty()` :

```
import pyeph

# Obtencion de la base de datos
eph = pyeph.obtener(data="eph", ano=2019, periodo=1, tipo_base='individual') # INGLÉS: ↵
↵ eph = pyeph.get(data="eph", year=2019, period=1, base_type='individual')

#Obtencion de la canasta
canasta = pyeph.obtener(data="canastas") # INGLÉS: canasta = pyeph.get(data="canastas")
```

(continué en la próxima página)

(proviene de la página anterior)

```
# Instanciar la clase Pobreza() o Poverty()
pobreza = pyeph.Pobreza(eph=eph, canasta=canasta) # INGLÉS: pobreza = pyeph.
↳Poverty(eph=eph, basket=canasta)
```

Una vez instanciada la clase con una EPH individual particular, se puede calcular la tasa de pobreza e indigencia de personas y de hogares, ejecutando los diferentes métodos.

Pobreza e Indigencia de Personas

Una vez instanciada la clase `pyeph.Pobreza()` o `pyeph.Poverty()` se ejecuta el método `poblacion()/population()`:

```
# Calculo de Pobreza e Indigencia (general)

print('Pobreza de personas:')
pobreza_personas = pobreza.poblacion() # INGLÉS: pobreza_personas = pobreza.population()

# Calculo de Pobreza e Indigencia agrupado por categoría (puede ser mas de una)
print('Pobreza de personas según sexo:')
pobreza_personas = pobreza.poblacion(agrupar_por='CH04') # INGLÉS: pobreza_personas =
↳pobreza.population(group_by='CH04')
```

Pobreza e Indigencia de Hogares

Una vez instanciada la clase `pyeph.Pobreza()` o `pyeph.Poverty()` se ejecuta el método `hogares()/household()`:

```
# Calculo de Pobreza e Indigencia (general)

print('Pobreza de hogares:')
pobreza_personas = pobreza.hogares() # INGLÉS: pobreza_personas = pobreza.household()
```

7.2 Cálculos de Mercado Laboral

La clase `pyeph.MercadoLaboral()` o `pyeph.LaborMarket()` - prepara las bases para los cálculos de los diferentes indicadores habituales vinculados al mercado laboral. Tiene tres métodos principales -`empleo()/employment()`, `desempleo()/unemployment()` y `actividad()/activity()`- que devuelven un dataframe con información referida a cada indicador.

7.2.1 Uso

La clase `pyeph.MercadoLaboral()` o `pyeph.LaborMarket()` recibe los siguientes parámetros:

```
pyeph.MercadoLaboral(eph)

o

pyeph.LaborMarket(eph)
```

7.2.2 Parámetros

Parámetros	Tipo de dato	Descripción
eph	pandas.DataFrame	Base de EPH individual publicada por INDEC.

7.2.3 Métodos

La clase `pyeph.MercadoLaboral()` o `pyeph.LaborMarket()` prepara las bases para los cálculos de los diferentes indicadores habituales vinculados al mercado laboral. Para obtener el dataframe con los resultados podrá consultar los métodos `empleo()/employment()`, `desempleo()/unemployment()` o `actividad()/activity()`, según corresponda.

- Empleo

Este método retorna como resultado un `pandas.DataFrame` donde se especifica [...].

```
pyeph.MercadoLaboral(eph).empleo(agrupar_por, div_por)

o

pyeph.LaborMarket(eph).employment(group_by, div_by)
```

Parámetros	Tipo de dato	Descripción
group_by (agrupar_por)	str or list, default: None	Nombre de la variable o lista de nombres de las variables por las cuales se desea agrupar y calcular los indicadores. Ejemplo: "CH04" o ["CH04", "REGION"]
div_by (div_por)	str, default="PT"	Indica la población que se utilizará para el cálculo de la tasa de empleo. Puede asumir "PT" indicando población total o "PET" indicando población en edad de trabajar.

- Desempleo

Este método retorna como resultado un `pandas.DataFrame` donde se especifica [...].

```
pyeph.MercadoLaboral(eph).desempleo(agrupar_por, div_por)

o

pyeph.LaborMarket(eph).unemployment(group_by, div_by)
```

Parámetros	Tipo de dato	Descripción
group_by (agrupar_por)	str or list, default: None	Nombre de la variable o lista de nombres de las variables por las cuales se desea agrupar y calcular los indicadores. Ejemplo: "CH04" o ["CH04", "REGION"]
div_by (div_por)	str, default="PET"	Indica la población que se utilizará para el cálculo de la tasa de empleo. Puede asumir "PT" indicando población total o "PET" indicando población en edad de trabajar.

- Actividad

Este método retorna como resultado un pandas.DataFrame donde se especifica [...].

```
pyeph.MercadoLaboral(eph).actividad(agrupar_por, div_por)

o

pyeph.LaborMarket(eph).activity(group_by, div_by)
```

Parámetros	Tipo de dato	Descripción
group_by (agrupar_por)	str or list, default: None	Nombre de la variable o lista de nombres de las variables por las cuales se desea agrupar y calcular los indicadores. Ejemplo: "CH04" o ["CH04", "REGION"]
div_by (div_por)	str, default="PT"	Indica la población que se utilizará para el cálculo de la tasa de empleo. Puede asumir "PT" indicando población total o "PET" indicando población en edad de trabajar.

7.2.4 Ejemplos

Primero se instancia la clase `pyeph.MercadoLaboral()` o `pyeph.LaborMarket()`:

```
import pyeph

# Obtencion de la base de datos
eph = pyeph.obtener(data="eph", ano=2019, periodo=1, tipo_base='individual') # INGLÉS:
↪ eph = pyeph.get(data="eph", year=2019, period=1, base_type='individual')
```

Resultado:

in- dex	CO- DU- SU	ANO 4	TRI- MES- TRE	FR- NRO 00	COM- PAR PO- NEN- TE	RE- GION	MAS 8	SAB- ME- RA- DO	108	PON- DE- RA	CH03	CH04	CH05	CH06	CH07	CH08	CH09	CH10	CH11	CH12
0	TQR-2019 MNOP- PUHKNKNC- DE- FOCD00629562	1	2	2	1	41	N	8	108	2	2	03/06289901			4	1	2	0	4	
1	TQR-2019 MNOP- PUHKNKNC- DE- FOCD00629562	1	2	3	1	41	N	8	108	3	2	29/12120055			4	1	1	1	4	
2	TQR-2019 MNOP- PUHKNKNC- DE- FOCD00629562	1	2	4	0	41	N	8	108	3	1	26/01120185			4	3	0	0	0	
3	TQR-2019 MNO- RUQHKNKNC- DE- FOCD00629563	1	1	1	1	41	N	8	108	1	2	30/03419785			4	1	2	0	6	
4	TQR-2019 MNO- RUQHKNKNC- DE- FOCD00629563	1	1	2	0	41	N	8	108	3	2	20/09920095			4	1	1	1	2	
5	TQR-2019 MNO- PUVH- KOKMC- DE- FOCD00629564	1	1	2	1	41	N	8	141	1	1	26/04519673			4	1	2	0	2	
6	TQR-2019 MNOQWVHMNLPC- DE- FOCD00629565	1	1	1	1	41	N	8	221	1	1	15/03689552			1	1	2	0	4	
7	TQR-2019 MNOQWVHMNLPC- DE- FOCD00629565	1	1	2	1	41	N	8	221	2	2	25/04629562			1	1	2	0	4	
8	TQR-2019 MNOQWVHMNLPC- DE- FOCD00629565	1	1	3	1	41	N	8	221	3	2	10/06219945			1	1	1	1	7	
9	TQR-2019 MNOQWXHMNLPC- DE- FOCD00629566	1	1	1	1	41	N	8	221	1	1	22/07719444			1	1	2	0	2	

```
# Instanciar la clase LaborMarket o MercadoLaboral
mercado_laboral = pyeph.MercadoLaboral(eph) # INGLÉS: mercado_laboral = pyeph.
↳ LaborMarket(eph)
```

Una vez instanciada la clase con una EPH individual particular, se pueden calcular los indicadores del mercado laboral, ejecutando los diferentes métodos.

Desempleo

Una vez instanciada la clase `pyeph.MercadoLaboral()` o `pyeph.LaborMarket()` se ejecuta el método `desempleo()/unemployment()`:

```
# Calculo de Desempleo General

print("\nTasa de desempleo considerando la Población en Edad de Trabajar:\n")
print(mercado_laboral.desempleo()) # INGLÉS: mercado_laboral.unemployment()
```

Resultado:

Tasa de desempleo considerando la Población en Edad de Trabajar:

Tasa de Desempleo
10.1

```
print("\nTasa de desempleo considerando la Población Total:\n")
print(mercado_laboral.desempleo(div_por='PT')) # INGLÉS: mercado_laboral.
↳ unemployment(div_by='PT')
```

Resultado:

Tasa de desempleo considerando la Población Total:

Tasa de Desempleo
10.1

```
# Calculo de Desempleo agrupado por categoría (puede ser mas de una)

print("\nTasa de desempleo considerando la Población en Edad de Trabajar según sexo:\n")
print(mercado_laboral.desempleo(agrupar_por = 'CH04')) # INGLÉS: mercado_laboral.
↳ unemployment(group_by='CH04')
```

Resultado:

Tasa de desempleo considerando la Población en Edad de Trabajar según sexo:

CH04	Tasa de Desempleo
1	9.2
2	11.2

```
print("\nTasa de desempleo considerando la Población Total según sexo:\n")
print(mercado_laboral.desempleo(agrupar_por = 'CH04', div_por='PT')) # INGLÉS: mercado_
↳ laboral.unemployment(group_by='CH04', div_by='PT')
```

Resultado:

Tasa de desempleo considerando la Población Total según sexo:

CH04	Tasa de Desempleo
1	9.2
2	11.2

Empleo

Una vez instanciada la clase `pyeph.MercadoLaboral()` o `pyeph.LaborMarket()` se ejecuta el método `empleo()/employment()`:

```
# Calculo de Empleo General

print("\nTasa de empleo considerando la Población en Edad de Trabajar:\n")
print(mercado_laboral.empleo()) # INGLÉS: mercado_laboral.employment()
```

Resultado:

Tasa de empleo considerando la Población en Edad de Trabajar:

Tasa de Empleo
42.3

```
print("\nTasa de empleo considerando la Población Total:\n")
print(mercado_laboral.empleo(div_por='PT')) # INGLÉS: mercado_laboral.employment(div_by=
↳ 'PT')
```

Resultado:

Tasa de empleo considerando la Población Total:

Tasa de Empleo
42.3

```
# Calculo de Empleo agrupado por categoría (puede ser mas de una)

print("\nTasa de empleo considerando la Población en Edad de Trabajar según sexo:\n")
```

(continúe en la próxima página)

(proviene de la página anterior)

```
print(mercado_laboral.empleo(agrupar_por = 'CH04')) # INGLÉS: mercado_laboral.
↳ employment(group_by='CH04')
```

Resultado:

Tasa de empleo considerando la Población en Edad de Trabajar según sexo:

CH04	Tasa de Empleo
1	49.7
2	35.3

```
print("\nTasa de empleo considerando la Población Total según sexo:\n")
print(mercado_laboral.empleo(agrupar_por = 'CH04', div_por='PT')) # INGLÉS: mercado_
↳ laboral.employment(group_by='CH04', div_by='PT')
```

Resultado:

Tasa de empleo considerando la Población Total según sexo:

CH04	Tasa de Empleo
1	49.7
2	35.3

Actividad

Una vez instanciada la clase `pyeph.MercadoLaboral()` o `pyeph.LaborMarket()` se ejecuta el método `actividad()/activity()`:

```
# Calculo de Actividad General

print("\nTasa de actividad considerando la Población en Edad de Trabajar:\n")
print(mercado_laboral.actividad()) # INGLÉS: mercado_laboral.activity()
```

Resultado:

Tasa de actividad considerando la Población en Edad de Trabajar:

Tasa de Actividad
47.0

```
print("\nTasa de actividad considerando la Población Total:\n")
print(mercado_laboral.actividad(div_por='PT')) # INGLÉS: mercado_laboral.activity(div_by=
↳ 'PT')
```

Resultado:

Tasa de actividad considerando la Población Total:

Tasa de Actividad
47.0

```
# Calculo de Actividad agrupado por categoría (puede ser mas de una)
print("\nTasa de actividad considerando la Población en Edad de Trabajar según sexo:\n")
print(mercado_laboral.actividad(agrupar_por = 'CH04')) # INGLÉS: mercado_laboral.
↪ activity(group_by='CH04')
```

Resultado:

Tasa de actividad considerando la Población en Edad de Trabajar según sexo:

CH04	Tasa de Actividad
1	54.7
2	39.8

```
print("\nTasa de actividad considerando la Población Total según sexo:\n")
print(mercado_laboral.actividad(agrupar_por = 'CH04', div_por='PT')) # INGLÉS: mercado_
↪ laboral.activity(group_by='CH04', div_by='PT')
```

Resultado:

Tasa de actividad considerando la Población Total según sexo:

CH04	Tasa de Actividad
1	54.7
2	39.8