



**Implementation in R**

**Methodologies and Data Science for Statistical Production**

**INE Chile**

**December 2022**

**Implementación en R**

**Implementation in R**

## Antes de comenzar...

Para seguir la demostración en tu computador, es necesario contar con un nivel básico de **R**

Si no sabes nada de R y quieres tener una idea general de la herramienta, no tendrás ningún problema

Es necesario tener un dominio muy básico de programación

## Before starting...

If you want to follow the demonstration on your machine, a basic knowledge of R is required

If you don't have any idea about R and you just want to get an overview, it is totally fine.

Very basic programming skill is required

# Demostración (Demonstration)

Lo primero, es descargar el paquete desde CRAN

The first step is to download the package from CRAN

```
install.packages("calidad")
```

... o versión en desarrollo desde [github](#)

... or developing version from [github](#)

```
devtools::install_github("inesscc/calidad")
```

Cargamos el paquete en la sesión y otras dependencias que usaremos

We load the package with other dependencies we will be using

```
library(calidad)  
library(survey)  
library(dplyr)
```

# Demostración (Demonstration)

Trabajaremos con los datos de la Encuesta de Caracterización Socioeconómica (CASEN) 2020 (cargados en el paquete)

We will work with the National Socioeconomic Survey (CASEN) 2020 (already loaded into the package)

Construyamos algunas variables necesarias para calcular indicadores relevantes del mercado laboral

Let's create some relevant variables for the labor market

```
casen_edit <- casen %>%  
  mutate(fdt = if_else(activ %in% c(1, 2), 1, 0, missing = 0), # fuerza de trabajo  
         ocupado = if_else(activ == 1, 1, 0, missing = 0), # persona ocupada  
         desocupado = if_else(activ == 2, 1, 0, missing = 0), # persona desocupada  
         metro = if_else(region == 13, 1, 0))
```

## Variables dummy

- fuerza de trabajo
- ocupado
- desocupado
- metro (pertenece a la región metropolitana)

## Dummy variables

- labour force
- employed
- unemployed
- metropolitan area

# Panorama general (Overview)

Declaramos el diseño complejo con la función `svydesign` de `survey`

We declare the complex design with the `svydesign` function from `survey`

```
dc <- svydesign(weights = ~expr, ids = ~cod_upm, strata = ~estrato, data = casen_edit )  
options(survey.lonely.psu = "certainty")
```

Debemos decirle a `R` qué hacer con la varianza cuando encuentra estratos con una sola UPM

We must tell `R` what to do with the variance when it finds strata with only one PSU

El paquete `calidad` tiene 2 tipos de funciones:

- `create_`: **crean** los insumos para el estándar
- `assess`: **evaluación** del estándar

Podemos hacer los siguientes cálculos

- media (`create_mean`)
- proporción o razón (`create_prop`)
- suma de variables continuas (`create_total`)
- conteo de unidades (`create_size`)

`calidad` has 2 types of functions

- `create_`: **to create** the quality indicators
- `assess`: **to assess** the quality according to some standard

The following estimations can be obtained

- mean (`create_mean`)
- proportion or ratio (`create_prop`)
- totals (`create_total`)
- size (`create_size`)

# Creando los insumos (getting the values)

Queremos calcular la edad media para mujeres y hombres

We want to get the mean age for women and men

```
create_mean(var = "edad", domains = "sexo", design = dc)
```

```
##      sexo      stat      se      df      n      cv
## 1      1 35.81776 0.1320879 10701 86096 0.003687776
## 2      2 38.88116 0.2030783 10818 99341 0.005223053
```

- `var`: variable a estimar
- `domains`: desagregaciones
- `design`: diseño muestral creado con `svydesign`

La función retorna:

- estimación (stat)
- error estándar (se)
- coeficiente de variación (CV)
- grados de libertad (df)
- tamaño muestral (n)

The function returns:

- estimation (stat)
- standard error (se)
- Coefficient of variation (CV)
- Degrees of freedom (df)
- Sample size (n)



# Creando los insumos (getting the values)

¿Y si queremos calcular la tasa de desempleo?

And if we would like to get the unemployment rate?

Para ello, contamos con la función `create_prop`

We have the `create_prop` function

```
create_prop(var = "desocupado", domains = "sexo", design = dc)
```

El problema es que el desempleo debe calcularse sobre una subpoblación específica (fuerza de trabajo)

We have one issue here: the unemployment must be obtained from a specific subpopulation (labor force)

Para ello, utilizamos el argumento `subpop`

we can use the `subpop` parameter

```
create_prop(var = "desocupado", domains = "sexo", subpop = "fdt", design = dc)
```

No olvidar que **subpop debe ser dummy**

Do not forget that **subpop must be a dummy variable**

Con `subpop` evitamos error en el cálculo de la

The use of `subpop` avoids errors in variance

# Creando los insumos (getting the values)

¿Qué pasa si queremos desagregar por más variables?

And what if we want to add more variables in domains?

Se debe agregar otra variable utilizando un signo +

That can be done using the "+" character

```
create_prop(var = "desocupado", domains = "sexo+metro", subpop = "fdt", design = dc)
```

##	sexo	metro	stat	se	df	n	cv
## 1	1	0	0.1171641	0.002540917	7977	34097	0.02168682
## 2	2	0	0.1417696	0.003012865	7750	27786	0.02125184
## 3	1	1	0.1097366	0.004349727	2019	9901	0.03963790
## 4	2	1	0.1364608	0.008565356	1972	9055	0.06276788

# Creando los insumos (getting the values)

Queremos calcular el número de ocupados  
respecto al número de ocupadas

$$\frac{\textit{SumaOcupadosHombre}}{\textit{SumaOcupadasMujer}}$$

We want to get the ratio between men employed  
and women employed

$$\frac{\textit{TotalEmployedMen}}{\textit{TotalEmployedWomen}}$$

Lo primero que debemos hacer es crear  
variables auxiliares

The first step is to create some new variables

```
casen_edit <- casen_edit %>%  
  mutate(ocupado_hombre = if_else(sexo == 1, ocupado, 0),  
         ocupada_mujer   = if_else(sexo == 2, ocupado, 0))
```

Volvemos a declarar el diseño para incluir las  
variables recién creadas

We have to declare the design again in order to  
include the new variables.

```
dc <- svydesign(weights = ~expr, ids = ~cod_upm, strata = ~estrato, data = casen_edit )
```

# Creando los insumos (getting the values)

La función `create_prop` permite incluir el argumento `denominator`

`create_prop` includes the parameter `denominator`

```
create_prop(var = "ocupado_hombre", denominator = "ocupada_mujer",
            subpop = "fdt", design = dc)
```

```
##      stat      se    df     n      cv
## 1 1.186844 0.0001418034 10590 80839 0.01003344
```

Podemos incluir el parámetro `domains`, si queremos desagregar

We can include `domains`, if we need it.

```
create_prop(var = "ocupado_hombre", denominator = "ocupada_mujer",
            domains = "metro", subpop = "fdt", design = dc)
```

```
##  metro      stat      se    df     n      cv
## 1     0 1.238095 0.01312396 8510 61883 0.01060013
## 2     1 1.127986 0.02073977 2080 18956 0.01838654
```

# Argumentos adicionales (other parameters)

Solo hemos revisado `create_prop` y `create_mean`

Todas las funciones del paquete operan de manera similar

Existen más argumentos

- `ci`
- `deff`
- `rel_error`
- ...

We took a look at `create_prop` and `create_mean`

All the functions works in a similar way

There are more parameters

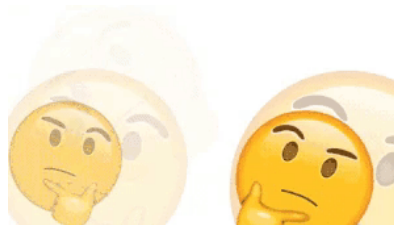
- `ci`
- `deff`
- `rel_error`
- ...

**Hasta el momento solo hemos visto la generacion de insumos**

**So far we have only learned how to create the inputs**

**¿Cómo podemos evaluar la calidad estadística?**

**How can we assess the statistical quality?**



# Evaluación de la calidad (quality assessment)

Evaluemos la calidad de la edad media, según sexo

Let's assess the quality for mean age by sex

```
est <- create_mean(var = "edad", domains = "sexo", design = dc)
assess(est)
```

```
##      sexo      stat      se      df      n      cv      eval_n      eval_df      eval_cv
## 1      1 35.81776 0.1320879 10701 86096 0.003687776 sufficient sample size sufficient df cv <= 0.15
## 2      2 38.88116 0.2030783 10818 99341 0.005223053 sufficient sample size sufficient df cv <= 0.15
##      label
## 1 reliable
## 2 reliable
```

Tenemos 4 columnas nuevas

- `eval_n`: indica si el tamaño muestral es suficiente
- `eval_df`: indica si los gl son suficientes
- `eval_cv`: indica el tramo en el que está el cv
- `label`: evaluación final de la estimación

We have 4 new columns

- `eval_n`: adequate sample size
- `eval_df`: adequate degrees of freedom
- `eval_cv`: adequate band for CV
- `label`: global assessment

Por defecto, las funciones de evaluación consideran el estándar ONE

By default the assessment function uses ONE Chile approach

# Evaluación de la calidad (quality assessment)

Veamos el caso de la tasa de desempleo

Let's move to the unemployment rate

```
est <- create_prop(var = "desocupado", subpop = "fdt", domains = "sexo", design = dc)
assess(est)
```

```
##      sexo      stat      se      df      n      cv      eval_n      eval_df prop_est
## 1      1 0.1138937 0.002402195 9996 43998 0.02109156 sufficient sample size sufficient df    <= 0.5
## 2      2 0.1393067 0.004271094 9722 36841 0.03065964 sufficient sample size sufficient df    <= 0.5
##      eval_type quadratic      eval_se eval_cv      label
## 1      Eval SE 0.02610701 admissible SE      <NA> reliable
## 2      Eval SE 0.02985879 admissible SE      <NA> reliable
```

La función agrega las siguientes columnas

The function adds the following columns

- `prop_est`
- `eval_type`
- `quadratic`
- `eval_se`
- `eval_cv`



# Evaluación de la calidad (quality assessment)

El estándar establece que un tabulado puede ser publicado si el 50% de sus celdas es fiable

ONE Chile stands that a chart can be published if 50% of its cells are reliable

Para saber si el tabulado debe ser publicado, usamos el argumento `publish`

To know if a chart can be published, we can use the `publish` parameter

```
est <- create_size(var = "desocupado", subpop = "fdt", domains = "region+sexo", design = dc)
assess(est, publish = T) %>%
  select(region, sexo, stat, label, publication, pass) %>%
  slice(1:4)
```

##	region	sexo	stat	label	publication	pass
## 1	1	1	9436	reliable	publish 100% reliable estimates	
## 2	2	1	21139	reliable	publish 100% reliable estimates	
## 3	3	1	8586	reliable	publish 100% reliable estimates	
## 4	4	1	22801	reliable	publish 100% reliable estimates	

Tenemos 2 nuevas columnas

We have 2 new columns

- `publication`: evaluación general del tabulado
- `pass`: porcentaje de celdas con categoría fiable

- `publication`: global assessment for a chart
- `pass`: percent cells with reliable label

# Evaluación de la calidad (quality assessment)



Podemos generar una visualización sencilla mediante la función `create_html`

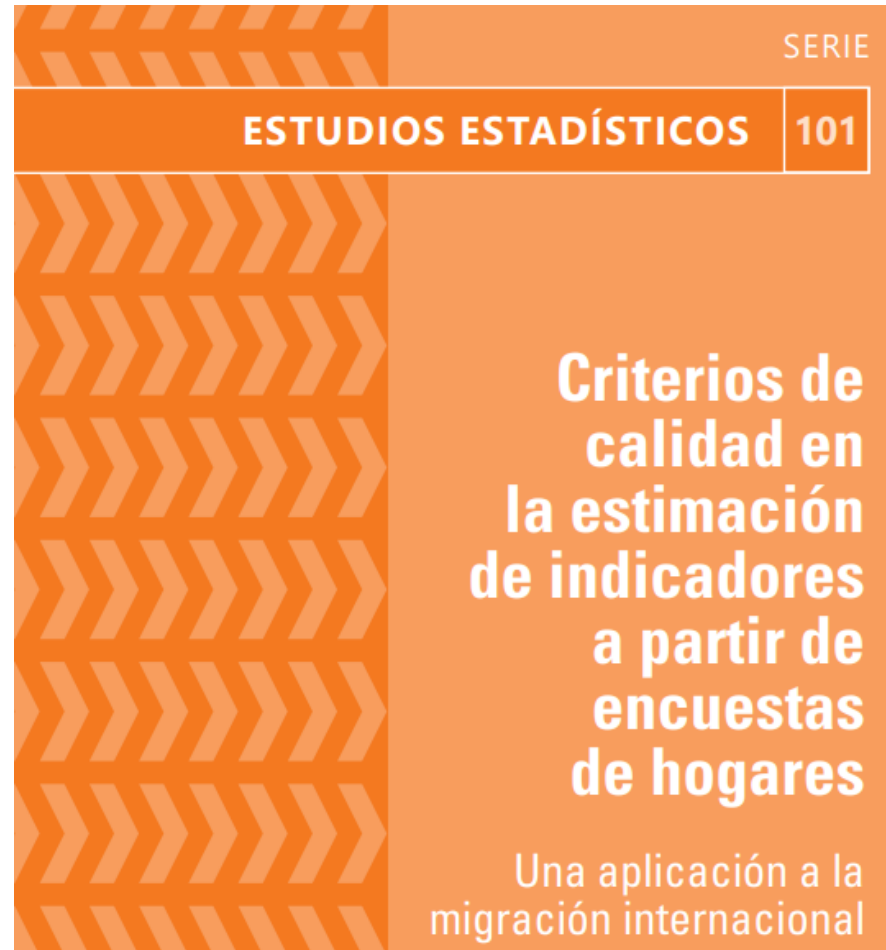
We can get a simple visualization using `create_html`

```
create_size(var = "desocupado", subpop = "fdt", domains = "region+sexo+pobreza", design = dc) %>%  
  assess( publish = T) %>%  
  slice(1:3) %>%  
  create_html( )
```

region	sexo	pobreza	stat	se	df	n	cv	eval_n	eval_df	eval_cv	label	publication	pass
1	1	1	1.075	236,75	20	27	0,22	insufficient sample size	sufficient df	cv between 0.15 and 0.3	non-reliable	do not publish	38.54% reliable estimates
2	1	1	1.633	444,91	13	20	0,27	insufficient sample size	sufficient df	cv between 0.15 and 0.3	non-reliable	do not publish	38.54% reliable estimates
3	1	1	632	185,09	9	15	0,29	insufficient sample size	sufficient df	cv between 0.15 and 0.3	non-reliable	do not publish	38.54% reliable estimates

**Estándar de calidad  
CEPAL**

**ECLAC criteria**



- **El estándar CEPAL considera:**

- coeficiente de variación
- **coeficiente de variación logarítmico**
- tamaño de muestra
- **tamaño de muestra efectivo**
- **conteo de casos no ponderado**
- grados de libertad

- **El estándar CEPAL considera:**

- coefficient of variation
- **logarithmic coefficient of variation**
- sample size
- **effective sample size**
- **non-weighted units**
- degrees of freedom

¡Veamos un poco de código!

Let's see some code!

# Estándar de calidad CEPAL (CEPAL approach)

Se deben incluir nuevos parámetros en las funciones `create_`

New parameters must be included in `create_`

```
est <- create_size(var = "desocupado", domains = "region+sexo", design = dc,  
  unweighted = T, deff = T, ess = T, df_type = "eclac")
```

Y agregar eclac en scheme

We have to add eclac in scheme

```
assess(est, scheme = "eclac") %>%  
  select(region, sexo, stat, n, df, cv, unweighted, ess, label) %>%  
  slice(1:6)
```

##	region	sexo	stat	n	df	cv	unweighted	ess	label
## 1	1	1	9436	3981	419	0.08412127	220	2545.121	publish
## 2	2	1	21139	3572	493	0.08197950	243	2130.673	publish
## 3	3	1	8586	3468	557	0.08745271	205	2076.391	publish
## 4	4	1	22801	3783	495	0.08145155	238	2402.469	publish
## 5	5	1	56607	8397	1119	0.06976443	511	2987.021	publish
## 6	6	1	24507	5830	611	0.07226518	327	3470.302	publish

# Estándar de calidad CEPAL (CEPAL approach)

Contamos con un atajo para retornar los indicadores de CEPAL

We can take a shortcut for eclac indicators

```
est <- create_size(var = "desocupado", domains = "region+sexo",  
                  design = dc, df_type = "eclac", eclac_input = TRUE)
```

# Umbrales flexibles (flexible thresholds)

¿Y si necesito fijar mis propios umbrales?

What if I need to set my own thresholds?



```
est <- create_size(var = "desocupado", domains = "region+sexo", design = dc,
  unweighted = T, deff = T, ess = T, df_type = "ine")
```

```
assess(est, scheme = "eclac", unweighted = 220, ess = 200) %>%
  select(region, sexo, stat, n, df, cv, unweighted, ess, label) %>%
  slice(1:6)
```

##	region	sexo	stat	n	df	cv	unweighted	ess	label
## 1	1	1	9436	220	144	0.08412127	220	140.6497	supress
## 2	2	1	21139	243	167	0.08197950	243	144.9478	supress
## 3	3	1	8586	205	138	0.08745271	205	122.7394	supress
## 4	4	1	22801	238	146	0.08145155	238	151.1466	supress
## 5	5	1	56607	511	322	0.06976443	511	181.7754	supress
## 6	6	1	24507	327	175	0.07226518	327	194.6464	supress



# Utilización de loops (loops)

Queremos calcular la media para varias variables

We want to get the mean for many variables

En este caso, queremos la media de `edad` y `ing_aut_hog`, según sexo

In this case, we want to get the mean for age and household\_income by sex

```
insumos <- data.frame()
for (v in c("edad", "ing_aut_hog")) {
  insumo <- create_mean(var = v, domains = "sexo", design = dc, rm.na = T )
  insumos <- bind_rows(insumos, insumo)
}
```

Podemos hacer lo mismo, utilizando el paquete `purrr` (mucho más recomendado que un for)

We can do exactly the same using a functional approach from `purrr` package (much better option than for loop)

```
insumos <- map_df(c("edad", "ing_aut_hog"), ~create_mean(var = .x, domains = "sexo",
  design = dc, rm.na = T ))
```

El paquete `survey` funciona con evaluaciones no estándar

Esto es muy útil porque permite una sintaxis compacta, pero requiere que el usuario conozca un poco el funcionamiento de R

`survey` package works with Non Standard Evaluations (NSE)

This is very useful because it allows for a compact syntax, but requires the user to know about how R works

```
svyby(~edad, design = dc, by = ~sexo, FUN = svymean)
```

Veamos cómo crear una función con el paquete `calidad`

Let's see how to create a function using `calidad`

```
create_and_assess <- function(var, dom, dc) {  
  create_mean(var, domains = dom, design = dc ) %>%  
    assess() %>%  
    select(dom, stat, n, df)  
}  
create_and_assess("edad", "sexo", dc)
```

```
##   sexo    stat      n    df  
## 1     1 35.81776 86096 10701  
## 2     2 38.88116 99341 10818
```

# ¿En qué estamos? (What are we doing now?)

- Mantenimiento constante
  - Junto a CEPAL estamos preparando instancias de difusión:
    - RTC
    - UNSD (aquí estamos)
  - Preparación de material de difusión
  - Integración de [calidad](#) con [inedata](#)
  - Comienzo de nuevos desarrollos
- Constant maintenance
  - Together with ECLAC we are preparing dissemination instances:
    - PSTN
    - UNSD (here we are)
  - Preparation of dissemination material
  - Integration of [calidad](#) with [inedata](#)
  - Start of new developments

## Open source development

El paquete [calidad](#) es un desarrollo completamente *open source*

En este [repositorio de github](#) pueden proponer nuevos desarrollos

Klaus Lehmann y Ricardo Pizarro son los mantenedores

Pueden generar *issues* o nuevas ramas de desarrollo

The [calidad](#) package is a completely *open source* development

In this [github repository](#) you can propose new developments

The maintainers are Klaus Lehmann y Ricardo Pizarro

You can create issues or new dev branches

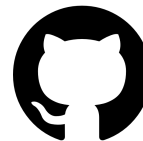
## Open source development

Si tienen **propuestas de mejora, reportes de errores o nuevos desarrollos**, estaremos felices de revisarlo e incorporarlo al paquete

If you have **improvement proposals, bug reports or new developments**, we will be happy to review it and incorporate it into the package



- Klaus Lehmann: kilehmannm@ine.gob.cl
- Ignacio Agloni: ifaglonij@ine.gob.cl
- Ricardo Pizarro: rapizarros@ine.gob.cl



<https://github.com/inesscc/calidad>



**Implementation in R**

**Methodologies and Data Science for Statistical Production**

**INE Chile**

**December 2022**