



# Taller R – Rifa Valentina

## Estimación desde muestras complejas

Nicolás Ratto

2022/02/02

# Contenidos

- Manipulación avanzada de data frames (funciones `pivot` y combinación de data frames)
- Manipulación de fechas: paquete `lubridate`
- Visualización de datos con `ggplot2`
- **Estimar desde diseños muestrales complejos (paquete `survey` y `srvyr`)**
- Hacer funciones, procesos iterativos e introducción a paquetes en R

# Muestras complejas en R

Inferencia a la población

# El desafío de la inferencia

La reducción de costos y esfuerzos que implica estudiar una población mediante una muestra, se compensa con el costo de la "incertidumbre" o "imprecisión".

La estadística nos da elementos para conocer y manejar esta incertidumbre.

Desde nuestra muestra vamos a estimar o inferir un valor aproximado del parámetro de la población.

Al hacer este proceso, no solo ocuparemos estimaciones puntuales (como medias, quantiles, medianas, etc.)

También tendremos que calcular la precisión de estas estimaciones

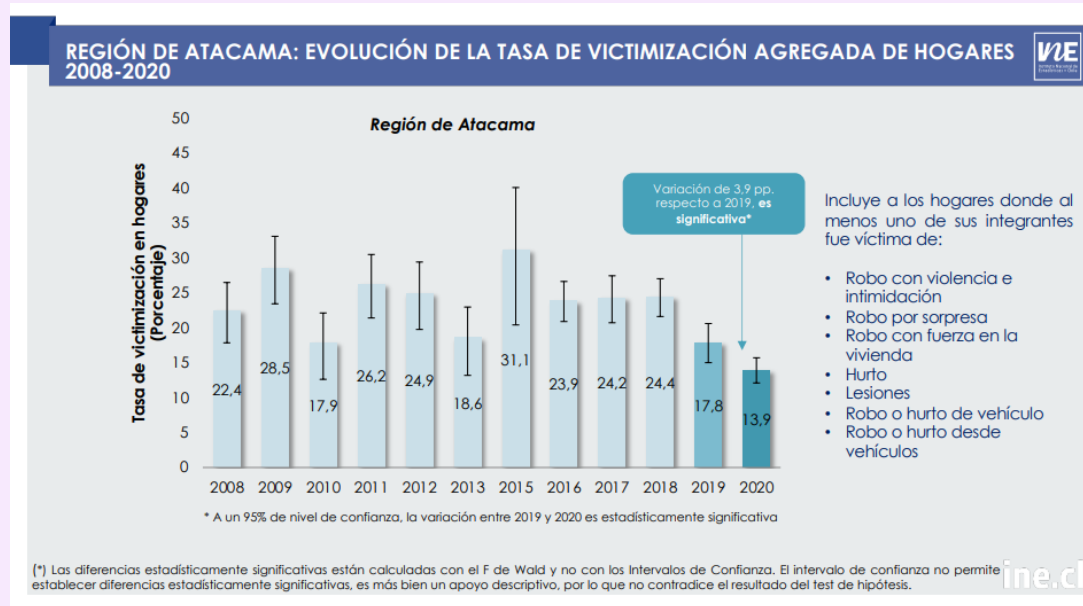
Todo estimador está compuesto por dos elementos

- Estimación puntual
- Precisión (ci, se, var, cv)

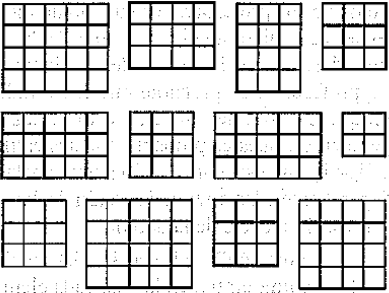
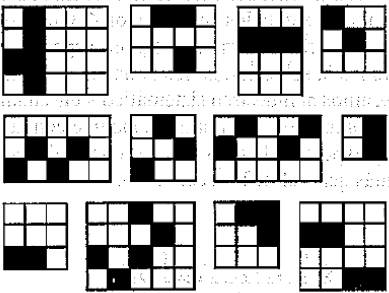
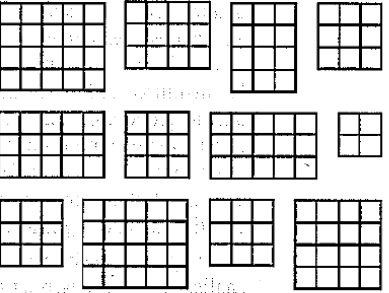
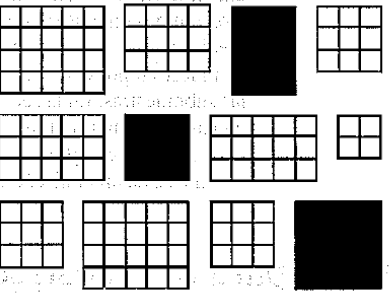
# Ej. Reporte de estimación



# Ej. Reporte de estimación



# Muestreo influye en inferencia

Muestreo estratificado	Muestreo por conglomerados
Cada elemento de la población está exactamente en un estrato.	Cada elemento de la población está en un solo conglomerado.
<p>Población de <math>H</math> estratos; el estrato <math>h</math> tiene <math>n_h</math> elementos:</p>  <p>Se extrae una muestra aleatoria simple de cada estrato:</p> 	<p>Muestreo por conglomerados en una etapa: población de <math>N</math> conglomerados:</p>  <p>Se extrae una muestra aleatoria simple de conglomerados; observe que todos los elementos dentro de los cúmulos están en la muestra:</p> 

# La forma incorrecta

Para estimar el valor del parámetro poblacional se requieren dos pasos:

- definir un estadístico como estimación del parámetro poblacional (estimación puntual)
- establecer en torno a un estadístico un intervalo de confianza para estimar en términos probabilísticos el parámetro.

Lo apropiado es reportar ambas cosas, pero es difícil cuando tenemos un **diseño muestral complejo**.

Para la estimación puntual, solo necesitaremos el peso de cada unidad de nuestros datos (weight)

Este **ponderador** o **factor de expansión (FE)** indica a cuántas unidades representa cada elemento de la muestra.

Los pesos **no** dan información acerca de la forma de determinar los errores estándar, siendo centrales para calcular los IC (precisión).

**Desv Est.** tiende a aumentar con conglomerados y a disminuir con estratos...



# Abramos R

Una vez más, trabajaremos con la Encuesta Nacional de Empleo.

¿Cuántas personas ocupadas existían en Chile en trimestre Ene-Mar 2021? [Consultar acá](#)

8.148.210 ocupados: 4.826.060 hombres (59,2%) y 3.322.150 mujeres (40,8%).

¿Como podemos reproducir este resultado desde la base de datos?

```
# Descargar la base de datos
ene <- read.csv(file = "https://www.ine.cl/docs/default-source/ocupacion/ene-mar-2021.csv")
```

```
ene %>% filter(activ==1) %>%
  group_by(sexo) %>% summarise(ocup=n()) %>% mutate(prop=ocup/sum(ocup))
```

```
## # A tibble: 2 x 3
##   sexo  ocup  prop
##   <int> <int> <dbl>
## 1     1 19443 0.569
## 2     2 14716 0.431
```

# Abramos R

ENE es un diseño por conglomerados, bietápico y estratificado.

Para estimar **puntualmente** de manera correcta necesitamos los **factores de expansión**

¿Como se comporta esta variable?

```
summary(ene$fact_cal)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	8.106	85.573	136.782	216.799	239.963	4107.542

¿Cuánto debería sumar?

```
## [1] 19595837
```

# Abramos R

Ahora filtramos para dejar solo los ocupados y agrupamos por sexo.

```
ene %>% filter(activ==1) %>%  
  group_by(sexo) %>%  
  summarise(ocupados=sum(fact_cal))
```

```
## # A tibble: 2 x 2  
##   sexo ocupados  
##   <int>   <dbl>  
## 1     1  1 4826056.  
## 2     2  2 3322150.
```

Y el total de ocupados

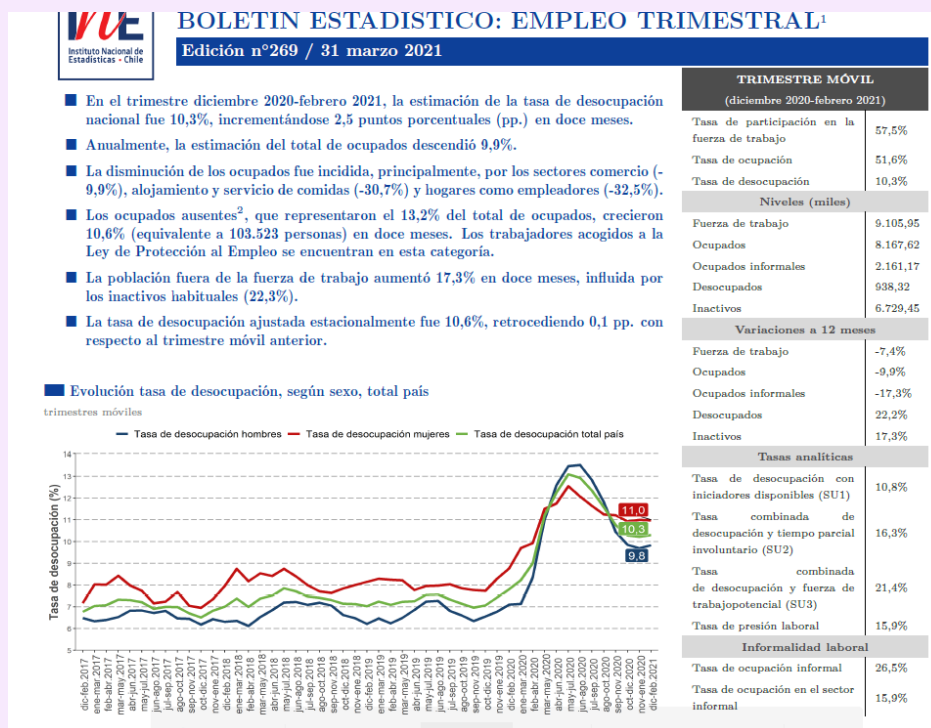
```
sum(ene[ene$activ==1,]$fact_cal, na.rm = TRUE)
```

```
## [1] 8148206
```

# Uso de estimaciones puntuales

Si bien es una mala práctica, tiene buen rendimiento y se usa en estudios descriptivos.

El mismo INE presenta las estimaciones puntuales sin advertencia de su precisión\*.



\*En los tabulados se señala cuando las estimaciones son poco o nada fiables

# Medir la precisión de la estimación

Si solo nos interesa la estimación puntual, podríamos simplemente usar el peso de cada caso y olvidarnos del resto.

Sin embargo, debemos ser capaces de conocer la precisión de nuestras estimaciones y poder determinar, al menos, si son significativamente diferentes de cero.

Para esto debemos suponer cosas, conocer la error estándar de nuestra variable, el nivel de confianza con el que estamos trabajando y otros elementos del diseño muestral.

# Medir la precisión de la estimación

Si la ENE fuera un simple Muestreo Aleatorio Simple, el intervalo en torno a las proporciones de hombres y mujeres se podría obtener definiendo:

$n = 90.387$ ,  $z = 1.96$ ,  $(pxq): 0.4691383 \times 0.5308617$

```
p <- as.vector(prop.table(table(ene$sexo)))[1]
n <- nrow(ene)
z <- 1.96 # para 95% de confianza
```

```
ci_p <- z * (sqrt(p*(1-p)/n))
(p-ci_p)*100 ; p*100 ; (p+ci_p)*100
```

```
## [1] 46.58848
```

```
## [1] 46.91383
```

```
## [1] 47.23917
```

Sin embargo, es un diseño complejo. Estimamos mal todo.

# La forma correcta: survey y srvyr

Para trabajar con muestras complejas en R son necesarios dos paquetes:

survey

srvyr



El primero fue creado por [Thomas Lumley](#).

El segundo es su adaptación por terceros para que dialogue con la gramática de `dplyr` y los pipes.

```
## Crear objeto tbl_svy
ds <- data %>%
  as_survey_design(ids = conglomerados, # ids=1 (no hay conglomerado)
                  strata = estratos,
                  weights = pesos)
```

# Diseño ENE

Muestreo bietápico -> conglomerados -> vivienda

- Unidad primaria de muestreo (UPM) definida como un conglomerado homogéneo, en términos de la cantidad de viviendas particulares que los componen. 35.140 upm
- Unidad secundaria de muestreo (USM) que corresponde a viviendas particulares ocupadas dentro de las UPM seleccionadas. 6.145.493 viviendas

La muestra de la ENE fue seleccionada desde el **Marco Muestral de Viviendas (MMV)** elaborado con Censo 2017.

El MMV contiene variables que permiten la clasificación de las UPM en:

- area rural / urbana
- comuna
- Tres niveles socioeconómicos

Se seleccionan 39.000 viviendas en 5.000 UPM aprox.



# La forma correcta: survey y srvyr

Todo este diseño se contiene en variables de la base de datos

```
ene %>% select(idrph,id_identificacion,estrato,  
               conglomerado,fact_cal) %>% head()
```

##	idrph	id_identificacion	estrato	conglomerado	fact_cal
## 1	544011	74938	5121	5.80100e+12	295.1434
## 2	72737011	74939	5121	5.80100e+12	261.4711
## 3	736011	74939	5121	5.80100e+12	270.1339
## 4	719011	74941	5121	5.80100e+12	295.1434
## 5	15805111	48044	10100111	1.01011e+09	126.3139
## 6	15804111	48044	10100111	1.01011e+09	108.6247

- Primer paso es cargar los paquetes

```
library(survey)  
library(srvyr)
```

# survey y srvyr

- Lo segundo es crear objeto `survey` con la ENE

Este objeto será del tipo `lista` en R y lo utilizaremos como si fuese la data frame

Todas las recodificaciones y ediciones hacerlas antes de crear el objeto `survey`.

Las variables con las que se harán agrupamientos deben mutarse a formato `factor`.

```
ene$activ<-as.factor(ene$activ) ## ojo  
ene$sexo<-as.factor(ene$sexo) ## ojo  
ene$categoria_ocupacion<-as.factor(ene$categoria_ocupacion) ## ojo
```

Crear el objeto `survey`

```
ds <- ene %>% as_survey_design(ids = conglomerado,  
                               strata = estrato,  
                               weights = fact_cal)
```

# survey y srvyr

Podemos agregar la corrección población finita para cada estrato con el argumento `fpc`

Esto aumenta la precisión de las estimaciones, pero no lo haremos acá para no complicar más las cosas (ver anexo al final con código)

- Lo tercero es definir algunas opciones generales

Cuando solo hay un conglomerado en un estrato, R no consigue calcular la varianza y arroja error (no se puede calcular varianza con una sola unidad).

Para solucionarlo indicamos que estos conglomerados solitarios no aporten a la varianza.

```
options(survey.lonely.psu = "certainty" )
```

# survey y srvyr

Y tenemos nuestro objeto survey llamado ds

```
class(ds) ## Consultar tipo de objeto
```

```
## [1] "tbl_svy"          "survey.design2" "survey.design"
```

- Hagamos nuestra primera estimación: número de personas por actividad

```
# Ocupados - Desocupados - Fuera de la FT
ds %>%
  group_by(activ) %>%
  summarise(trabajadores=survey_total(na.rm = TRUE))
```

```
## # A tibble: 4 x 3
##   activ trabajadores trabajadores_se
##   <fct>         <dbl>         <dbl>
## 1 1           8148206.         91044.
## 2 2           941088.         26958.
## 3 3           6763752.         70928.
## 4 <NA>       3742791.         62359.
```

# survey y srvyr

Ya no solo aparece la estimación puntual, se agrega medida de precisión.

Error estándar indica la variabilidad de las medias muestrales.

Como la desviación estándar de la población rara vez se conoce, el error estándar de la media suele estimarse como la desviación estándar de la muestra dividida por la raíz cuadrada del tamaño de la muestra.

Con el error estandar podemos obtener los intervalos de confianza

$$\left[ \bar{x} + z_{\alpha/2} \frac{sd}{\sqrt{n}}, \bar{x} - z_{\alpha/2} \frac{sd}{\sqrt{n}} \right]$$

¿Hay que seguir haciendo cálculos a mano?

No, survey lo hace por nosotros...

# survey y srvyr

```
ds %>% group_by(activ) %>%  
  summarise(trabajadores=survey_total(na.rm = TRUE,  
                                       vartype=c("ci")))
```

```
## # A tibble: 4 x 4  
##   activ trabajadores trabajadores_low trabajadores_upp  
##   <fct>         <dbl>         <dbl>         <dbl>  
## 1 1           8148206.         7969723.         8326688.  
## 2 2           941088.          888240.          993936.  
## 3 3          6763752.         6624705.         6902799.  
## 4 <NA>        3742791.         3620543.         3865039.
```

Por defecto survey trabaja con nivel de confianza del 95% ( $z=1,96$ ).

# survey y srvyr

Podemos cambiar el nivel de confianza también. Por ejemplo al 90%

```
ds %>% group_by(activ) %>%  
  summarise(trabajadores=survey_total(na.rm = TRUE,  
                                       vartype=c("ci"),level=c(0.90)))
```

```
## # A tibble: 4 x 4  
##   activ trabajadores trabajadores_low trabajadores_upp  
##   <fct>          <dbl>          <dbl>          <dbl>  
## 1 1            8148206.          7998426.          8297985.  
## 2 2             941088.           896739.           985437.  
## 3 3           6763752.          6647066.          6880438.  
## 4 <NA>         3742791.          3640203.          3845379.
```

O al 99% (mucha confianza, poca precisión)

```
## # A tibble: 4 x 4  
##   activ trabajadores trabajadores_low trabajadores_upp  
##   <fct>          <dbl>          <dbl>          <dbl>  
## 1 1            8148206.          7913610.          8382801.  
## 2 2             941088.           871625.          1010551.  
## 3 3           6763752.          6580990.          6946515.
```

# survey y srvyr

Con esto, la tasa de desocupación publicada de 10,4% tendrá incertidumbre.

```
tasa<-ds %>% group_by(activ) %>%  
  summarise(trabajadores=survey_total(na.rm = TRUE,  
                                       vartype=c("ci")))
```

```
tasa <- tasa %>%  
  filter(activ==1|activ==2) %>% # seleccionar ocup y desocup  
  janitor::adorn_totals("row") # total por columna
```

Tasas de desocupación:

```
tasa[2,2:4]/tasa[3,2:4]
```

```
##   trabajadores trabajadores_low trabajadores_upp  
## 1      0.1035381          0.1002759          0.1066384
```

El error es mínimo (+0,3%) por el alto número de observaciones (37.589).



# survey y srvyr

¿Cuál sería la tasa de desocupación en la submuestra de las personas de la Región de la Araucanía? (4.728 personas en la muestra)

```
tasa<-ds %>% filter(region==9) %>% group_by(activ) %>%  
  summarise(trabajadores=survey_total(na.rm = TRUE,  
                                       vartype=c("ci"))) %>%  
  filter(activ==1|activ==2) %>% # seleccionar ocup y desocup  
  janitor::adorn_totals("row")
```

```
##   activ trabajadores trabajadores_low trabajadores_upp  
##     1      361231.6      329543.66      392919.55  
##     2       30028.1       22096.23      37959.97  
## Total      391259.7      351639.89      430879.53
```

```
##   trabajadores trabajadores_low trabajadores_upp  
## 1  0.07674724      0.06283767      0.08809881
```

Intervalo de +- 1,3%, ya comienza a ser más relevante.

(imaginen el error para una comuna o un grupo específico)

# survey y srvyr

Podemos también estimar los totales de una variable, no solo de las unidades.

Horas semanales trabajadas en el país

```
ds %>%  
  summarise(c2_1_3= survey_total(c2_1_3,na.rm = TRUE,vartype=c("ci")))
```

```
##           c2_1_3 c2_1_3_low c2_1_3_upp  
## 1 371732324    359201972    384262677
```

Teniendo el NT

```
ds %>% filter(activ==1) %>%  
  summarise(nt= survey_total(na.rm = TRUE,vartype=c("ci")))
```

```
##           nt  nt_low  nt_upp  
## 1 8148206 7969722 8326689
```

Sabemos que en promedio se trabajan 45,6 horas semanales (+- 0,6 horas).

# survey y srvyr

No solo podemos estimar totales, también proporciones, medias, etc.

## Proporciones por categoría de respuesta

```
ds %>% filter(categoria_ocupacion!=0) %>%  
  group_by(categoria_ocupacion) %>%  
  summarise(proportion = survey_prop(vartype = c("ci"),na.rm = TRUE)) %:
```

categoria_ocupacion	proportion	proportion_low	proportion_upp
1	0.0303037	0.0276027	0.0330046
2	0.2020540	0.1949819	0.2091261
3	0.5984368	0.5895705	0.6073031
4	0.1352569	0.1293014	0.1412124
5	0.0211308	0.0187257	0.0235360
6	0.0039637	0.0027804	0.0051471
7	0.0088541	0.0073189	0.0103892

# survey y srvyr

¿Y que tanto cambian las proporciones sin diseño complejo?

Con survey

Sin survey

categoria_ocupacion	proportion	proportion_se
1	0.0303037	0.0013778
2	0.2020540	0.0036075
3	0.5984368	0.0045227
4	0.1352569	0.0030379
5	0.0211308	0.0012268
6	0.0039637	0.0006036
7	0.0088541	0.0007831

prop
0.0344272
0.2125355
0.5612576
0.1534881
0.0238297
0.0030446
0.0114172

# survey y srvyr

## Media

Función `survey_mean`, incluyendo variable numérica

```
ds %>% filter(categoria_ocupacion!=0) %>% group_by(categoria_ocupacion)
  summarise(media_edad = survey_mean(edad,vartype = c("ci"),na.rm = TRUE)
```

```
## # A tibble: 7 x 4
##   categoria_ocupacion media_edad media_edad_low media_edad_upp
##   <fct>              <dbl>         <dbl>         <dbl>
## 1 1                    50.4           49.2           51.5
## 2 2                    45.8           45.3           46.3
## 3 3                    40.2           40.0           40.5
## 4 4                    42.2           41.7           42.7
## 5 5                    48.4           47.1           49.6
## 6 6                    50.1           45.6           54.6
## 7 7                    40.8           38.4           43.1
```

# survey y srvyr

## Media

Horas semanales promedio en actividad principal

```
ds %>% filter(categoria_ocupacion!=0) %>% group_by(categoria_ocupacion)
  summarise(c2_1_3 = survey_mean(c2_1_3,vartype = c("ci"),na.rm = TRUE)).
```

```
## # A tibble: 7 x 4
##   categoria_ocupacion c2_1_3 c2_1_3_low c2_1_3_upp
##   <fct>              <dbl>    <dbl>    <dbl>
## 1 1                48.6      46.7      50.6
## 2 2                36.8      34.5      39.1
## 3 3                48.6      46.9      50.3
## 4 4                47.5      44.6      50.4
## 5 5                30.9      29.4      32.4
## 6 6                47.5      44.7      50.3
## 7 7                40.0      36.3      43.7
```

# Mediana (2do cuartil)

```
ds %>% filter(categoria_ocupacion!=0) %>%  
  group_by(categoria_ocupacion) %>%  
  summarise(c2_1_3 = survey_median(c2_1_3,vartype = c("ci"),na.rm = TRUE)
```

```
## # A tibble: 7 x 4  
##   categoria_ocupacion c2_1_3 c2_1_3_low c2_1_3_upp  
##   <fct>              <dbl>    <dbl>    <dbl>  
## 1 1                45        45        45  
## 2 2                30        30        35  
## 3 3                45        45        45  
## 4 4                44        44        44  
## 5 5                35        28        40  
## 6 6                45        45        48  
## 7 7                40        35        42
```

# survey y srvyr

También se pueden calcular:

## Cuartiles (y otros percentiles)

```
ds %>% filter(!is.na(activ)) %>% group_by(activ) %>%  
  summarise(edad=survey_quantile(edad,c(0.25, 0.5, 0.75),na.rm = TRUE))
```

## Desviación estándar y varianza

```
ds %>% group_by(categoria_ocupacion) %>%  
  summarise(sd_edad=survey_sd(edad,na.rm = TRUE),  
            varianza_edad=survey_var(edad,na.rm = TRUE))
```

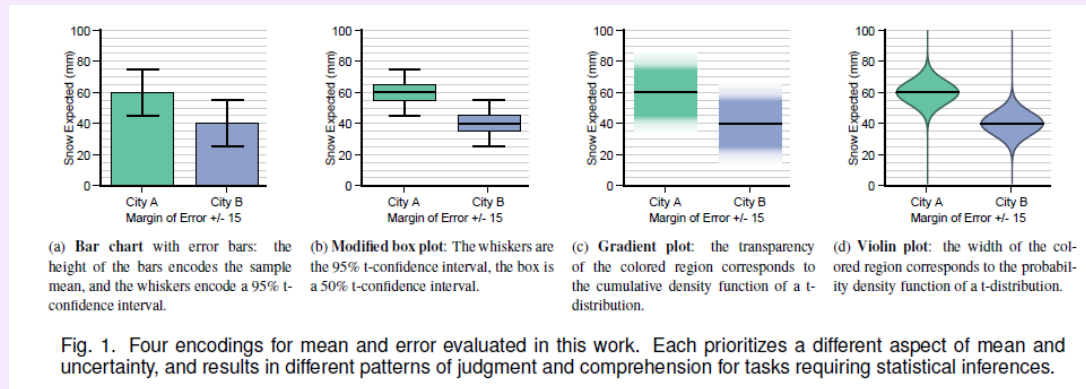


# Muestras complejas en R

Visualización de la incertidumbre

# Visualizar la incertidumbre

La forma más común es el gráfico de barras o de líneas con barras de error (A).



La visualización puede aportar más de un intervalo (B)

O incluso se puede ir más allá, visualizando la incertidumbre de forma continua (C y D)

La visualización puede confundir, presentándose EE como si fuesen intervalos

Si solo se reporta el EE como IC, se está utilizando un nivel de confianza del 68% ( $z=1$ ). Si no se explicita, se hace pasar como si fuese de 95% ( $z=1.96$ ).

IC al 95% es el doble que un IC al 68%.

# Visualizar la incertidumbre

## Gráfico de barras

Para visualizar datos utilizando diseños complejos, el "input" que agreguemos a `ggplot2` ya debe haber sido calculado con `survey`

Por ejemplo, trabajadores por categoría en la ocupación.

Primero creamos la tabla con la inferencia.

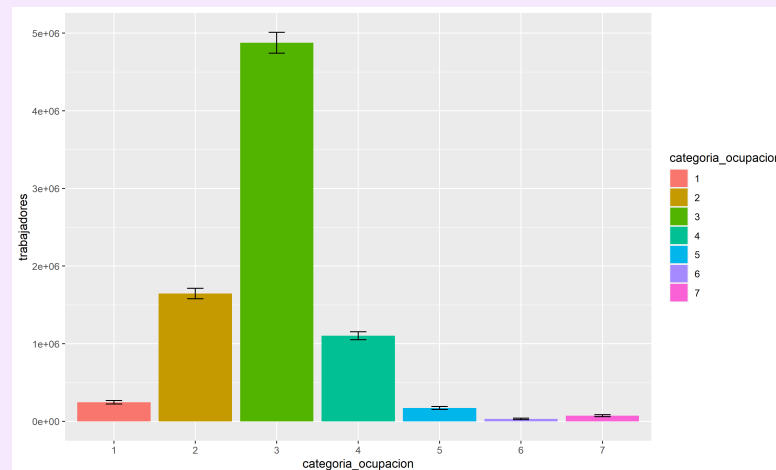
```
tabla1 <- ds %>% filter(categoria_ocupacion!=0) %>%  
  group_by(categoria_ocupacion) %>%  
  summarise(trabajadores=survey_total(na.rm = TRUE,  
                                       vartype=c("ci")))  
tabla1[1:3,] %>% kable()
```

categoria_ocupacion	trabajadores	trabajadores_low	trabajadores_upp
1	246920.4	224238.9	269601.8
2	1646377.5	1578675.8	1714079.2

# Visualizar la incertidumbre

## Gráfico de barras

```
tabla1 %>%  
  ggplot(aes(x=categoria_ocupacion,  
             y=trabajadores,  
             fill=categoria_ocupacion))+  
  geom_bar(stat = "identity") +  
  geom_errorbar(aes(ymin=trabajadores_low, ymax=trabajadores_upp),  
               width=0.2, position=position_dodge(.9))
```



# Visualizar la incertidumbre

## Gráfico de líneas

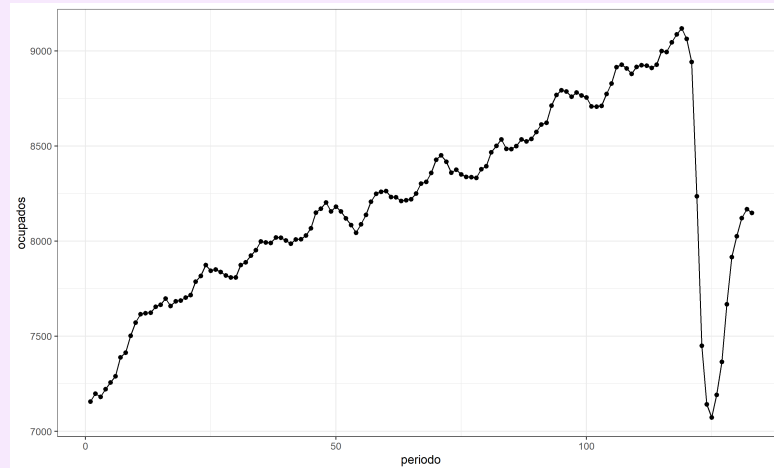
Data con la evolución de los ocupados (en miles) en Chile entre 2010 y 2021.

```
## # A tibble: 133 x 4
##   ocupados periodo ocupados_low ocupados_upp
##   <dbl>    <int>      <dbl>      <dbl>
## 1    7156.         1    6798.    7514.
## 2    7199.         2    6839.    7559.
## 3    7182.         3    6823.    7541.
## 4    7222.         4    6860.    7583.
## 5    7257.         5    6894.    7619.
## 6    7289.         6    6925.    7654.
## 7    7389.         7    7020.    7759.
## 8    7414.         8    7044.    7785.
## 9    7503.         9    7128.    7878.
## 10   7572.        10    7194.    7951.
## # ... with 123 more rows
```

# Visualizar la incertidumbre

## Gráfico de líneas sin IC

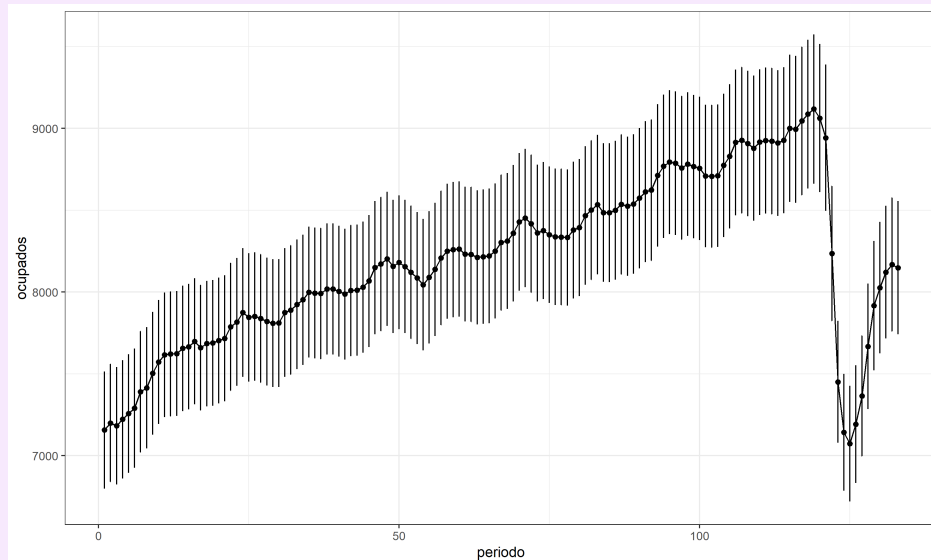
```
serie %>% ggplot(aes(x=periodo, y=ocupados)) +  
  geom_line() +  
  geom_point() +  
  theme_bw()
```



# Visualizar la incertidumbre

Gráfico de líneas, con medidas de precisión

```
serie %>% ggplot(aes(x=periodo, y=ocupados)) +  
  geom_line() +  
  geom_point() +  
  theme_bw() +  
  geom_errorbar(aes(ymin=ocupados_low, ymax=ocupados_upp), width=.01)
```



# Correlación con pesos

```
## Correlación considerando pesos
library(weights)
ene2<-ene %>% select(edad,c2_1_1,c2_2_1)
weighted_corr<-wtd.cor(ene2, weight = ene$fact_cal)
weighted_corr$correlation
```

```
##          edad      c2_1_1      c2_2_1
## edad      1.00000000 -0.03761003  0.1405247
## c2_1_1    -0.03761003  1.00000000  0.2348693
## c2_2_1     0.14052466  0.23486931  1.0000000
```

```
# Correlación considerando sin considerar pesos
cor(ene2, use = "complete.obs")
```

```
##          edad      c2_1_1      c2_2_1
## edad      1.00000000 -0.05798551  0.04832614
## c2_1_1    -0.05798551  1.00000000  0.24264390
## c2_2_1     0.04832614  0.24264390  1.00000000
```



# Anexo 1. FPC

Corrección por población finita

```
## obtener población de cada estrato
ene <- ene %>% group_by(estrato) %>%
  mutate(pob_estrata=sum(fact_cal)) %>%
  ungroup()

## RE crear objeto survey incluyendo argumento fpc
ds2 <- ene %>% as_survey_design(ids = conglomerado,
                                strata = estrato,
                                weights = fact_cal,
                                fpc=pob_estrata)

ds2 %>%
  group_by(activ) %>%
  summarise(trabajadores=survey_total(na.rm = TRUE))
```

# Bibliografía y elementos consultados

Heiss, A. [Uncertainty](#). En curso "Data Visualization".

INE. Boletín Mensual DEF 2021 [Encuesta Nacional de Empleo](#).

[Xaringan: Presentation Ninja](#), de Yihui Xie. Para generar esta presentación.

[Lehmann et al \(2021\) Presentación paquete "calidad" en LatinR](#)

Revisar Correl, M. [Error bars considered harmful](#) para conocer la discusión