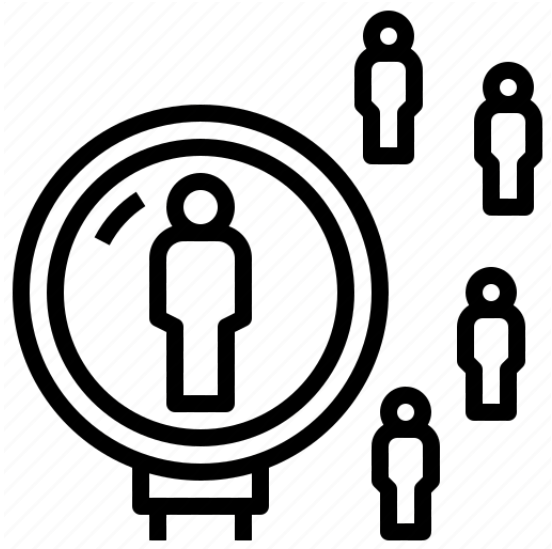
Universidad de la República
Facultad de Ciencias Economicas y de Administración
Licenciatura en Estadística

Muestreo II



Proyecto final

Ignacio Acosta - Valentina Caldiroli - Mauro Loprete

Parte 1 : Estimaciones con ponderadores originales

Se calculan las estimaciones con los ponderadores originales, estimaciones de la tasa de desempleo, la proporción de personas pobres e ingreso promedio.

Dada la existencia de no respuesta en la muestra y el tratamiento realizado, estamos frente a **una postura deterministica de la no respuesta**.

A continuación se muestra el código utilizado para realizar las diferentes estimaciones :

```
muestra %>%
    as_survey_design(
        ids = id_hogar,
        weight = w0,
        strata = estrato
    ) %T>%
    assign(
        "diseño",
        .,
        envir = .GlobalEnv
    ) %>%
    filter(
        R > 0
    ) %>%
    summarize(
        td = survey_ratio(
            desocupado,
            activo,
            deff = TRUE,
            vartype = c("se","cv")
        ),
        pobre = survey_mean(
            pobreza,
            deff = TRUE,
            vartype = c("se","cv")
        ),
        yprom = survey_mean(
            ingreso,
            deff = TRUE,
            vartype = c("se","cv")
        )
    ) %>%
    assign(
      "est_originales",
      .,
      envir = .GlobalEnv
    )
```

Los resultados se encuentran en el siguiente cuadro:

Cuadro 1: Estimaciones poblacionales usando ponderadores originales

Variable	Estimación puntual	Error estandar	CV	deff
pobre	0.085	0.004	0.047	2.976
td	0.082	0.003	0.041	1.079
yprom	22037.709	257.069	0.012	0.937

En base al cuadro, podemos ver que los errores estandar son relativamente chicos, de manera análoga podemos tomar el incremento de varianza respecto a un diseño simple, en base al $deff$, estos son altos debido al diseño en varias etapas de esta encuesta.

Tasa de no respuesta

Desde un enfoque determinístico de la no respuesta, podemos dividir a la muestra en aquellos individuos que respondierón y en aquellos que no lo hicierón.
Es decir, podemos particionar la muestra en aquellos respondentes r_u y no respondentes $s - r_u$.

Una medida de interés, es ver la proporción de respuestas en nuestra muestra, definida como :

$$p_{r_u} = \frac{n_{r_u}}{n_s}$$

Para nuestra muestra particular, este dato viene dado por :

```
muestra %>%
    summarize.(
        tr = mean(R)
    ) %>%
    mutate.(
        tnr = 1 - tr
    ) %>%
    assign(
        "tasaRespuesta",
        .,
        envir = .GlobalEnv
    )
```

Cuadro 2: Tasa de Respuesta

Tasa de Respuesta	Tasa de No Respuesta
0.54	0.46

En base a este indicador, podemos ver que poco más de la mitad de las personas seleccionadas en la muestra se pudo recabar información.

Por último, podemos ver la tasa de no respuesta poblacional, definida como :

$$\hat{p}_{r_u} = \frac{\sum_{r_u} w_0}{\sum_{r_s} w_0} = \frac{\hat{N}_{r_u}}{\hat{N}_s}$$

```
muestra %>%
    summarize.(
        tr = sum(R*w0) / sum(w0)
    ) %>%
    assign(
        "tasaRespuestapob",
        .,
        envir = .GlobalEnv
    )
```

Cuadro 3: Tasa de Respuesta poblacional

Tasa de respuesta poblacional	Tasa de no respuesta poblacional
0.54	0.46

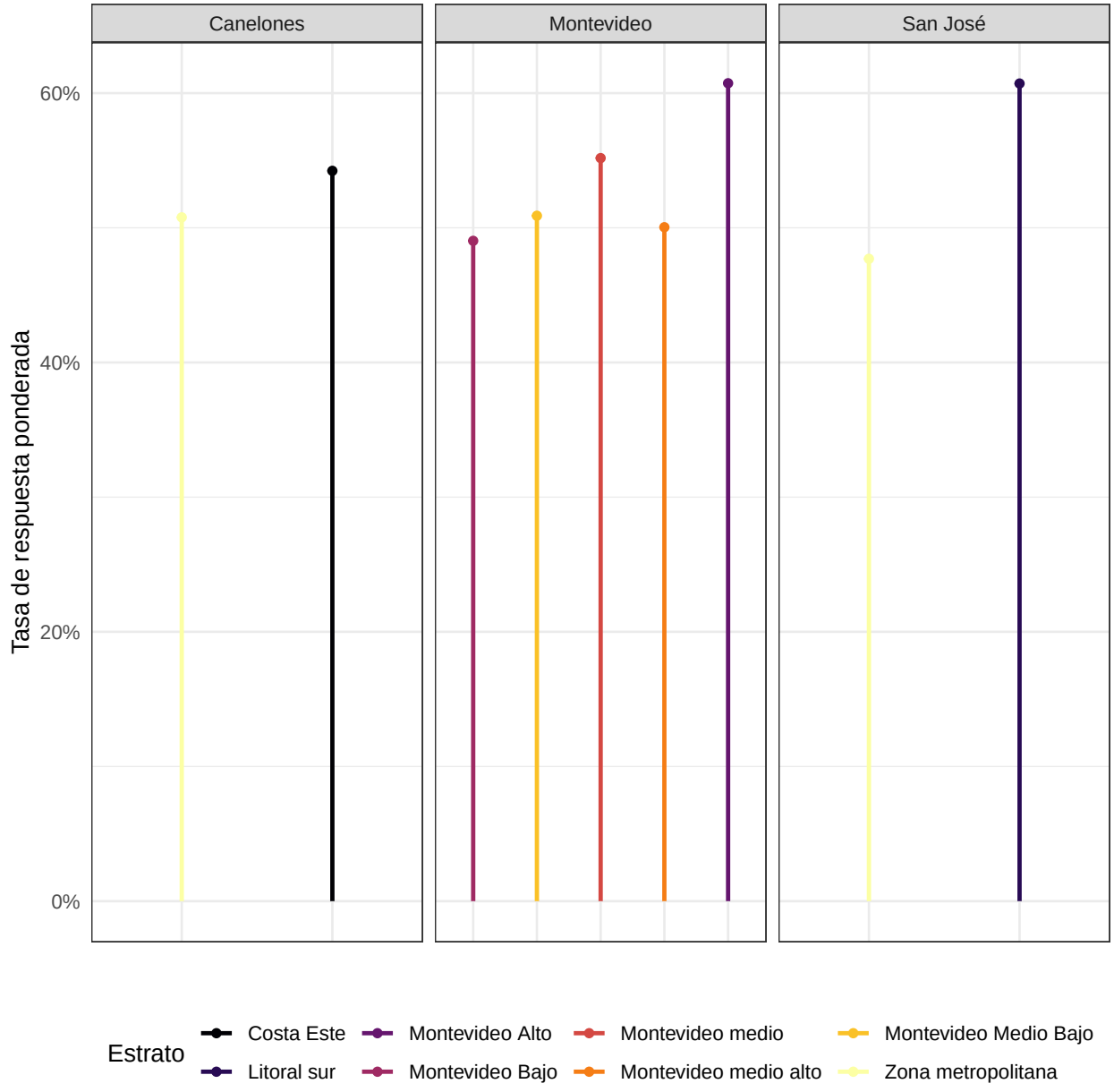
De manera análoga, la tasa de respuesta es identica a la anterior, si consideramos dos cifras significativas. Esta estimación puede tener la siguiente interpretación : *el porcentaje de la población que estoy cubriendo una vez que expanda la muestra*, que para este caso particular, **es sumamente bajo**.

Parte 2

Parte a

A continuación se calcuara la tasa de respuesta asumiento un patrón del tipo MAR, es decir, la no respuesta depende de covariables no utilizadas para la estimación.
Los diferentes grupos de no respuesta se construiran en base al departamento y estrato al que pertenecen, para aprovechar al máximo las variables consideradas en el marco.

A exepción de Montevideo, Canelones y San José, a cada departamento se le imputara la tasa de respuesta de su mismo departamento. Esto puede llegar a hacer diferencias ya que si resumimos esta variable considerando la interacción de estos dos grupos, se pueden encontrar diferentes comportamientos en la tasa de respuesta :



En base a la gráfica podemos ver diferentes comportamientos en los diferentes departamentos, aunque la mayor diferencia se nota en San José, con una tasa de respuesta mayor en el Litoral Sur respecto al de la zona metropolitana.

A lo que refiere a Montevideo, se puede ver una relación creciente (no monotona gracias al estrato medio alto) al estrato referido al contexto economico [1] para la tasa de respuesta, mientras que para Canelones no se notan grandes diferencias.

Una vez hecho esto, continuaremos con el ajuste por no respuesta para los ponderadores originales :

```
muestra %>%
    as_survey_design(
        ids = id_hogar,
        weight = w_nr_post,
        strata = estrato
    ) %T>%
    assign(
        "diseño",
        .,
        envir = .GlobalEnv
    ) %>%
    filter(
        R > 0
    ) %>%
    summarize(
        td = survey_ratio(
            desocupado,
            activo,
            deff = TRUE,
            vartype = c("se","cv")
        ),
        pobre = survey_mean(
            pobreza,
            deff = TRUE,
            vartype = c("se","cv")
        ),
        yprom = survey_mean(
            ingreso,
            deff = TRUE,
            vartype = c("se","cv")
        ),
        deffK = deff(
          w_nr_post,
          type = "kish"
        )
    ) %>%
    assign(
      "est_ponderados_nr",
      .,
      envir = .GlobalEnv
    )
```

[1] Asumiendo que los estratos son los mismos que el de la ECH

Resultando así, las estimaciones del punto anterior y considerando además el *Efecto diseño de Kish* que nos indica cuanto au

Cuadro 4: Estimaciones poblacionales usando ponderadores por no respuesta

Variable	Estimación puntual	Error estandar	CV	deff	Efecto diseño de Kish
pobre	0.088	0.004	0.047	3.068	1.03
td	0.082	0.003	0.042	1.097	1.03
yprom	21914.940	257.318	0.012	0.952	1.03

Referencias

Libros consultados

[26] Hadley Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016. ISBN: 978-3-319-24277-4. URL: https://ggplot2.tidyverse.org.

Paquetes de R

[1] Stefan Milton Bache y Hadley Wickham. magrittr: A Forward-Pipe Operator for R. R package version 2.0.1. 2020. URL: https://CRAN.R-project.org/package=magrittr.

[2] Andrew Bray y col. infer: Tidy Statistical Inference. R package version 1.0.0. 2021. URL: https://CRAN.R-project.org/package=infer.

[3] Tianqi Chen y col. xgboost: Extreme Gradient Boosting. R package version 1.5.0.2. 2021. URL: https://github.com/dmlc/xgboost.

[4] Mark Fairbanks. tidytable: Tidy Interface to data.table. R package version 0.6.5. 2021. URL: https://github.com/markfairbanks/tidytable.

[5] Greg Freedman Ellis y Ben Schneider. srvyr: dplyr-Like Syntax for Summary Statistics of Survey Data. R package version 1.1.0. 2021. URL: https://CRAN.R-project.org/package=srvyr.

[6] Lionel Henry y Hadley Wickham. purrr: Functional Programming Tools. R package version 0.3.4. 2020. URL: https://CRAN.R-project.org/package=purrr.

[7] Max Kuhn. modeldata: Data Sets Used Useful for Modeling Packages. R package version 0.1.1. 2021. URL: https://CRAN.R-project.org/package=modeldata.

[8] Max Kuhn. tune: Tidy Tuning Tools. R package version 0.1.6. 2021. URL: https://CRAN.R-project.org/package=tune.

[9] Max Kuhn. workflowsets: Create a Collection of tidymodels Workflows. R package version 0.1.0. 2021. URL: https://CRAN.R-project.org/package=workflowsets.

[10] Max Kuhn y Hannah Frick. dials: Tools for Creating Tuning Parameter Values. R package version 0.0.10. 2021. URL: https://CRAN.R-project.org/package=dials.

[11] Max Kuhn y Davis Vaughan. parsnip: A Common API to Modeling and Analysis Functions. R package version 0.1.7. 2021. URL: https://CRAN.R-project.org/package=parsnip.

[12] Max Kuhn y Davis Vaughan. yardstick: Tidy Characterizations of Model Performance. R package version 0.0.8. 2021. URL: https://CRAN.R-project.org/package=yardstick.

[13] Max Kuhn y Hadley Wickham. recipes: Preprocessing and Feature Engineering Steps for Modeling. R package version 0.1.17. 2021. URL: https://CRAN.R-project.org/package=recipes.

[14] Max Kuhn y Hadley Wickham. Tidymodels: a collection of packages for modeling and machine learning usi 2020. URL: https://www.tidymodels.org.

[15] Max Kuhn y Hadley Wickham. tidymodels: Easily Install and Load the Tidymodels Packages. R package version 0.1.4. 2021. URL: https://CRAN.R-project.org/package=tidymodels.

[16] Kirill Müller. here: A Simpler Way to Find Your Files. R package version 1.0.1. 2020. URL: https://CRAN.R-project.org/package=here.

[17] Kirill Müller y Hadley Wickham. tibble: Simple Data Frames. R package version 3.1.5. 2021. URL: https://CRAN.R-project.org/package=tibble.

[18] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: https://www.R-project.org/.

[19] Tyler Rinker y Dason Kurkiewicz. pacman: Package Management Tool. R package version 0.5.1. 2019. URL: https://github.com/trinker/pacman.

[20] Tyler W. Rinker y Dason Kurkiewicz. pacman: Package Management for R. version 0.5.0. Buffalo, New York, 2018. URL: http://github.com/trinker/pacman.

[21] David Robinson, Alex Hayes y Simon Couch. broom: Convert Statistical Objects into Tidy Tibbles. R package version 0.7.9. 2021. URL: https://CRAN.R-project.org/package=broom.

[22] Julia Silge y col. rsample: General Resampling Infrastructure. R package version 0.1.0. 2021. URL: https://CRAN.R-project.org/package=rsample.

[23] Richard Valliant, Jill A. Dever y Frauke Kreuter. PracTools: Tools for Designing and Weighting Survey Sam R package version 1.2.2. 2020. URL: https://CRAN.R-project.org/package=PracTools.

[24] Davis Vaughan. workflows: Modeling Workflows. R package version 0.2.4. 2021. URL: https://CRAN.R-project.org/package=workflows.

[25] Hadley Wickham. forcats: Tools for Working with Categorical Variables (Factors). R package version 0.5.1. 2021. URL: https://CRAN.R-project.org/package=forcats.

[27] Hadley Wickham. tidyr: Tidy Messy Data. R package version 1.1.4. 2021. URL: https://CRAN.R-project.org/package=tidyr.

[28] Hadley Wickham y Jennifer Bryan. readxl: Read Excel Files. R package version 1.3.1. 2019. URL: https://CRAN.R-project.org/package=readxl.

[29] Hadley Wickham y Dana Seidel. scales: Scale Functions for Visualization. R package version 1.1.1. 2020. URL: https://CRAN.R-project.org/package=scales.

[30] Hadley Wickham y col. dplyr: A Grammar of Data Manipulation. R package version 1.0.7. 2021. URL: https://CRAN.R-project.org/package=dplyr.

[31] Hadley Wickham y col. ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. R package version 3.3.5. 2021. URL: https://CRAN.R-project.org/package=ggplot2.

[32] Hao Zhu. kableExtra: Construct Complex Table with kable and Pipe Syntax. R package version 1.3.4. 2021. URL: https://CRAN.R-project.org/package=kableExtra.