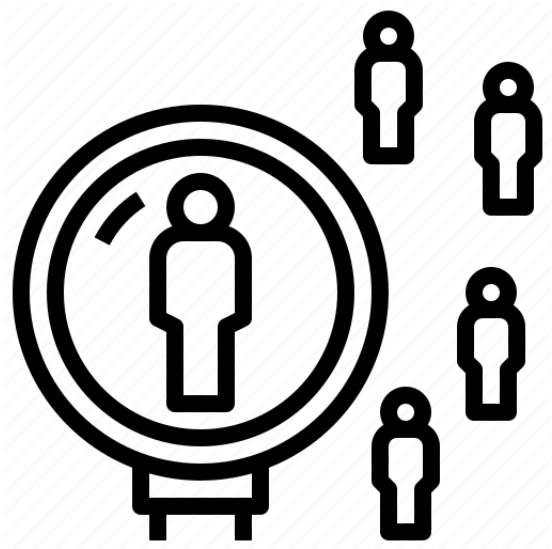
Universidad de la República
Facultad de Ciencias Economicas y de Administración
Licenciatura en Estadística

# 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 |

Con base en el cuadro, se puede ver que los errores estándar son relativamente chicos.
Analizando el incremento de varianza respecto a un diseño simple, haciendo uso del efecto $deff$, puede verse como las mismas son altas debido al diseño en varias etapas de esta encuesta.

### Tasa de no respuesta

Un enfoque determinista de la tasa de no respuesta plantea a la misma como el cociente entre la cantidad de personas que sí respondieron a la pregunta de interés y el total de personas de la muestra.
Es decir, es posible particionar la muestra en los respondentes $r_u$ y no respondentes $s - r_u$.

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

Para nuestra muestra particular, esta medida viene dada 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 |

Considerando 2 cifras significativas en el análisis, la tasa calculada en esta sección coincide con la anterior.

Esta estimación tiene la siguiente interpretación : *%54 es el porcentaje de la población que estoy cubriendo una vez expandida la muestra*, que para este caso particular, **es sumamente bajo**.

# Parte 2

## Parte a

A continuación se calcuara la tasa de respuesta asumiendo un patrón del tipo MAR. En el mismo se supone que la tasa de respuesta puede ser expresada como una función de un set de covariables, es decir $\Phi_i = \Phi_i(\vec{X})$.

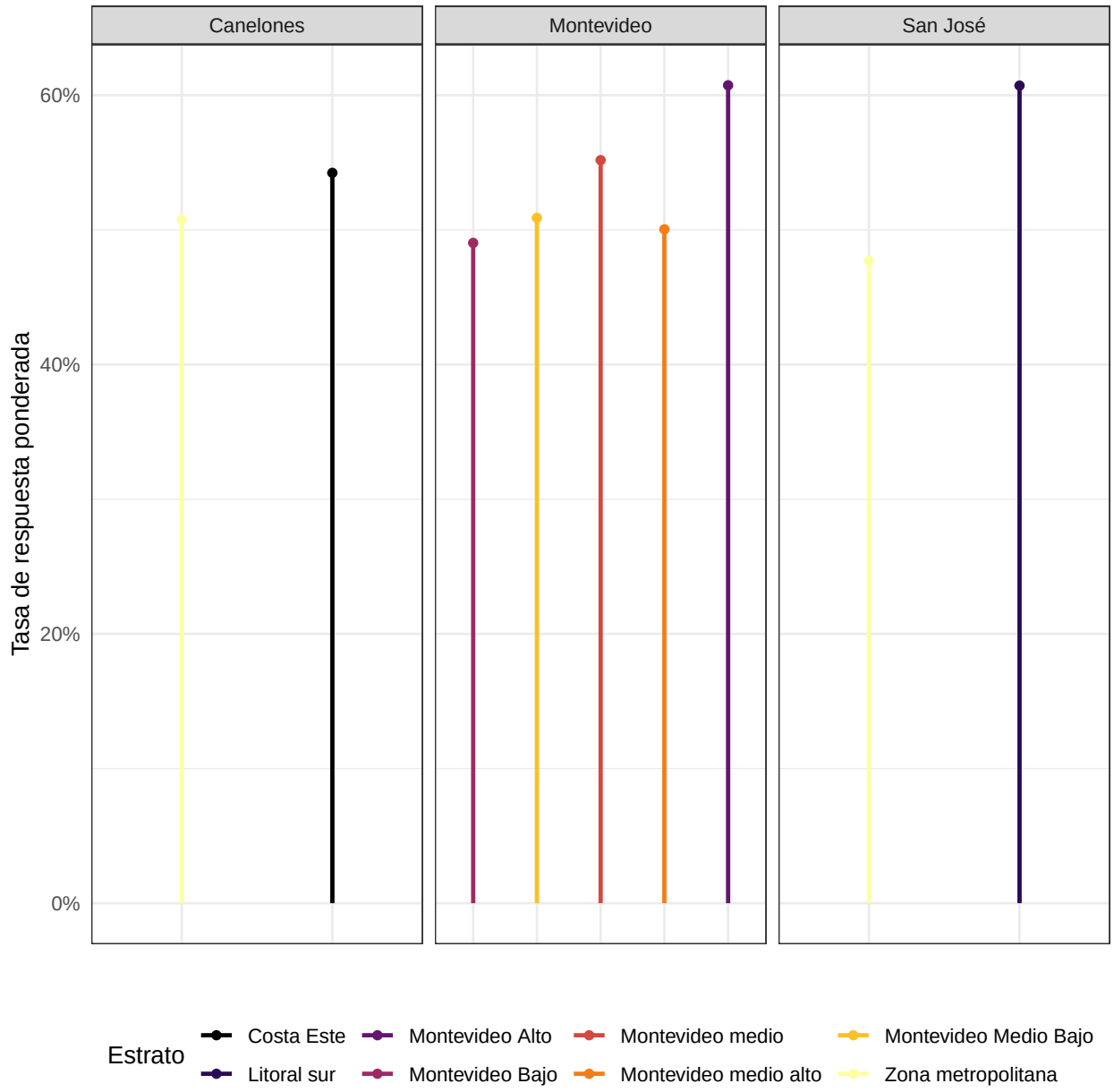
Este tipo de estrategia se basa en crear grupos de individuos con comportamiento similar en la no respuesta, a todos los ponderadores considerados dentro del mismo grupo se le aplicara el mismo ajuste $\Phi_i$.

Los diferentes grupos de no respuesta se construirán con base en el departamento y estrato al que pertenecen, para aprovechar al máximo las variables consideradas en el marco.

A exepción de Montevideo, Canelones y San José que presentan comportamientos disímiles entre algunos subsectores, a cada departamento se le imputara la tasa de respuesta de su mismo departamento.

```
muestra %>%
    summarize.(
        tr_w_estrato_dpto = weighted.mean(
            R
        ),
        .by = c(
            "estrato",
            "dpto"
        )
    ) %>%
    assign(
        "tr_estrato_dpto",
        .,
        envir = .GlobalEnv
    )

muestra %<>%
    left_join.(
        tr_estrato_dpto,
        by = c(
            "estrato" = "estrato",
            "dpto" = "dpto"
        )
    ) %>%
    mutate.(
        w_nr_post = w0 / tr_w_estrato_dpto
    )
```




La gráfica ilustra los diferentes comportamientos de la tasa de respuesta 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.

En lo que refiere a Montevideo,puedese puede ver una relación creciente (no monótona debido al estrato medio alto) al estrato referido al contexto económico [1] para la tasa de respuesta, mientras que para Canelones no se notan grandes diferencias.

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

Una vez hecho esto, se continuará 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
    )
```

Resultando así, las estimaciones del punto anterior y considerando además el *Efecto diseño de Kish*. Una vez realizado el ajuste, estimaciones de las variables, cálculo de error estándar, coeficiente de variación y efecto diseño, los mismos difirieron un poco con los calculados en el enfoque determinístico.

Algo que puede llamar la atención es como bajó el ingreso promedio, así como todas las estimaciones referidas a su estadístico.

**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 |

A lo que refiere al efecto diseño de Kish, podemos ver que se encuentra en un nivel de 1.03 un aumento poco considerable en la variabilidad de los ponderadores, respecto a un diseño autoponderado.

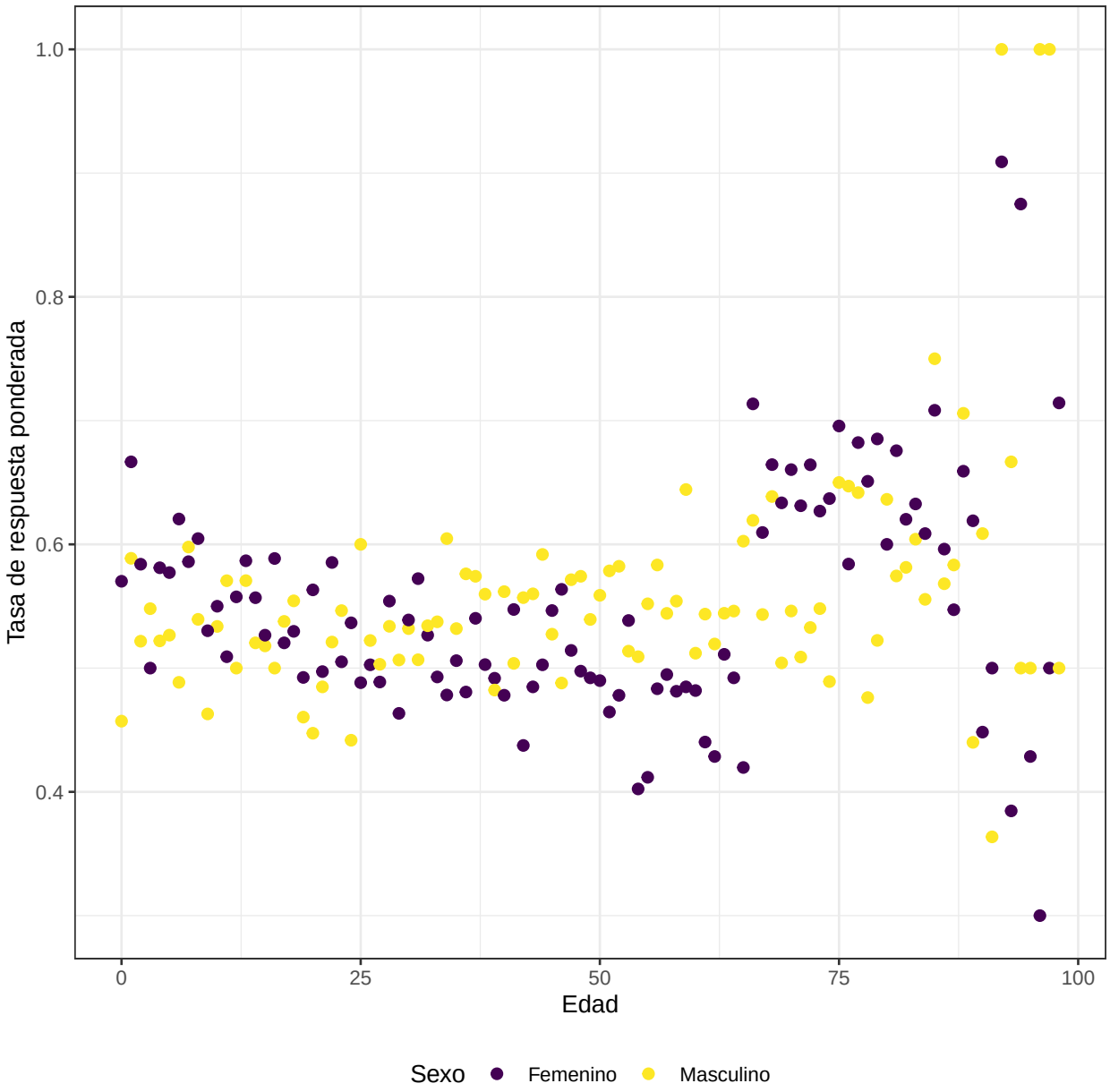
## Parte b

Mediante el uso de modelos de Machine Learning de aprendizaje supervizado, estimaremos el *propensity score*. Se hizo uso del método de **Gradient boosting**.

Este tipo de modelos es similar al Random Forest, generando diferentes arboles de decisión y promediando sus resultados.
La diferencia con este último es que la secuencia de árboles es dependiente de la realización anterior, ya que en cada paso se minimiza una función de pérdida (predicciones contra valores observados).

Este tipo de modelos supera a los del tipo de RF, ya que cada nuevo árbol de clasificación se genera tomando en cuenta los errores del paso anterior y no simplemente por azar.

Dado que este tipo de modelos tiende a tener problemas de sobreajuste, debemos de elegir una cantidad de árboles moderada, en nuestro caso 200.




Salvo exepciones, puede verse que en edades jovenes y adultas la tasa de respuesta no es diferente según sexo. En edades ancianas, la tasa de respuesta en mujeres tiende a aumentar respecto a la de los hombres, es por esto que sería bueno incluir estas dos variables en el modelo de clasificación.

```
boost_tree(
    trees = 300
) %>%
set_engine(
    "xgboost"
) %>%
set_mode(
    "classification"
) %>%
fit(
    as.factor(R) ~ estrato + sexo + edad + dpto, data = muestra
```

```
) %>%
assign(
    "modelo_boost",
    .,
    envir = .GlobalEnv
)

## [19:40:50] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0, the def
```

### Análisis de ajuste

```
## Warning in vec2table(truth = truth, estimate = estimate, dnn = dnn, ...): `truth`
was converted to a factor
```

**Cuadro 5:** Matriz de confusión

| Predicción | Observados | |
|---|---|---|
| | 0 | 1 |
| 0 | 7597 | 3947 |
| 1 | 4763 | 10562 |

A lo que refiere a la sensibilidad del modelo podemos ver que es de 70 %, mientras que la especificidad es de un casi 62 % y un error total de 31 % . Se calculan las propensiones individuales y ajustamos los ponderadores por no respuesta:

```
muestra %<>%
    bind_cols.(
        pred_boost
    ) %>%
    mutate.(
        w_nr_boost = (w0*R)/(pred_boost)
    )
```

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

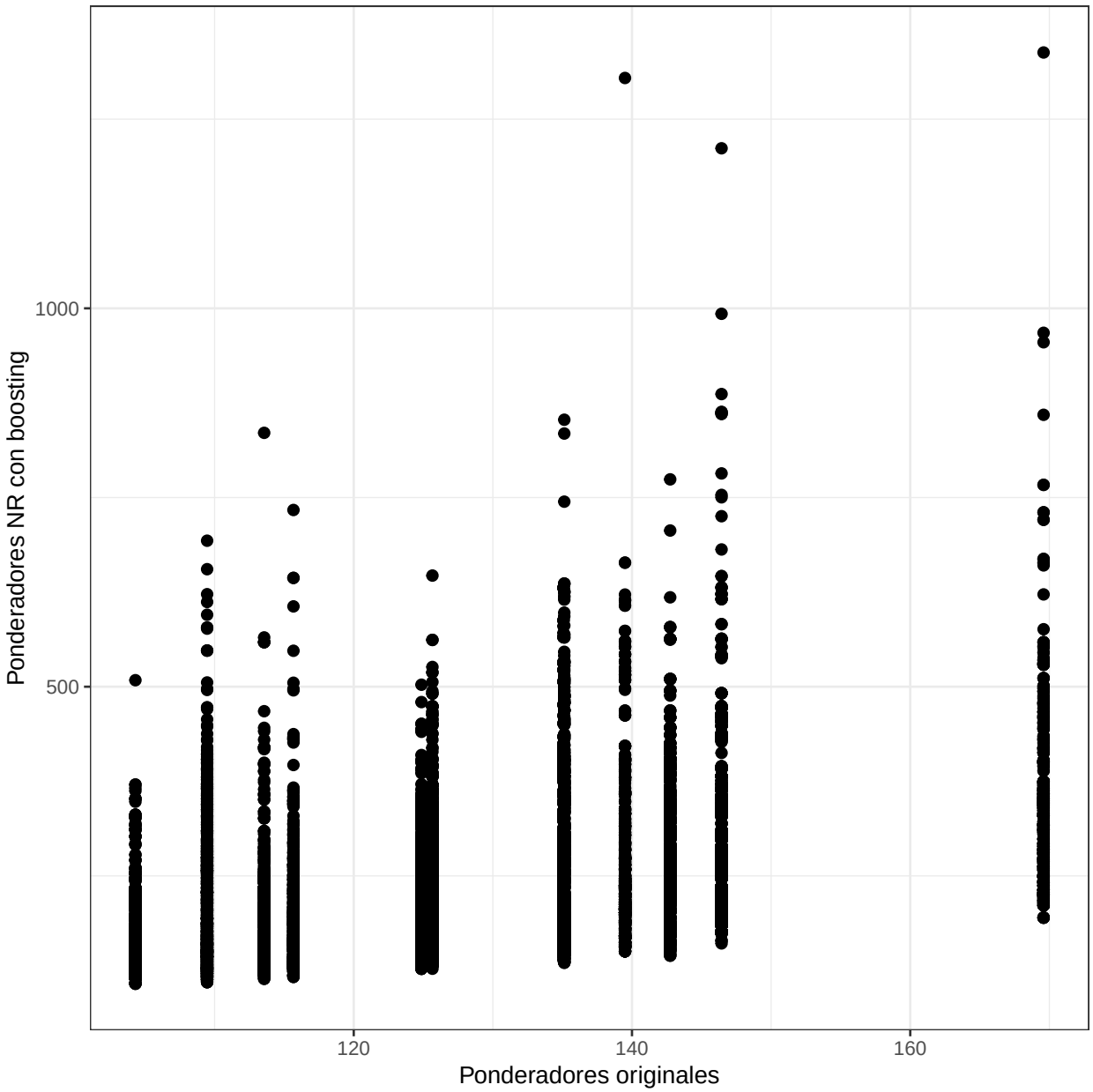
```
            vartype = c("se","cv")
        ),
        yprom = survey_mean(
            ingreso,
            deff = TRUE,
            vartype = c("se","cv")
        ),
        deffK = deff(
          w_nr_boost,
          type = "kish"
        )
    ) %>%
    assign(
      "est_ponderados_nr_boost",
      .,
      envir = .GlobalEnv
    )
```

**Cuadro 6:** Estimaciones poblacionales usando ponderadores por no respuesta utilizando Boosting

| Variable | Estimación puntual | Error estandar | CV | deff | Efecto diseño de Kish |
|---|---|---|---|---|---|
| pobre | 0.089 | 0.004 | 0.047 | 3.171 | 1.153 |
| td | 0.083 | 0.004 | 0.044 | 1.257 | 1.153 |
| yprom | 21878.361 | 268.478 | 0.012 | 1.039 | 1.153 |

Una vez realizadas las estimaciones, se puede ver que las estimaciones difieren poco. Se nota a su vez un leve aumento en los errores estándar y coeficientes de variación, sin embargo se puede notar un aumento en el efecto diseño, así como también en el efecto diseño de Kish.

Este último, si bien es mas alto que el anterior basándonos en la regla empírica, no es algo para preocuparse por el momento.
Por último, veremos un gráfico de dispersión de los ponderadores originales, respecto a los recién ajustados.



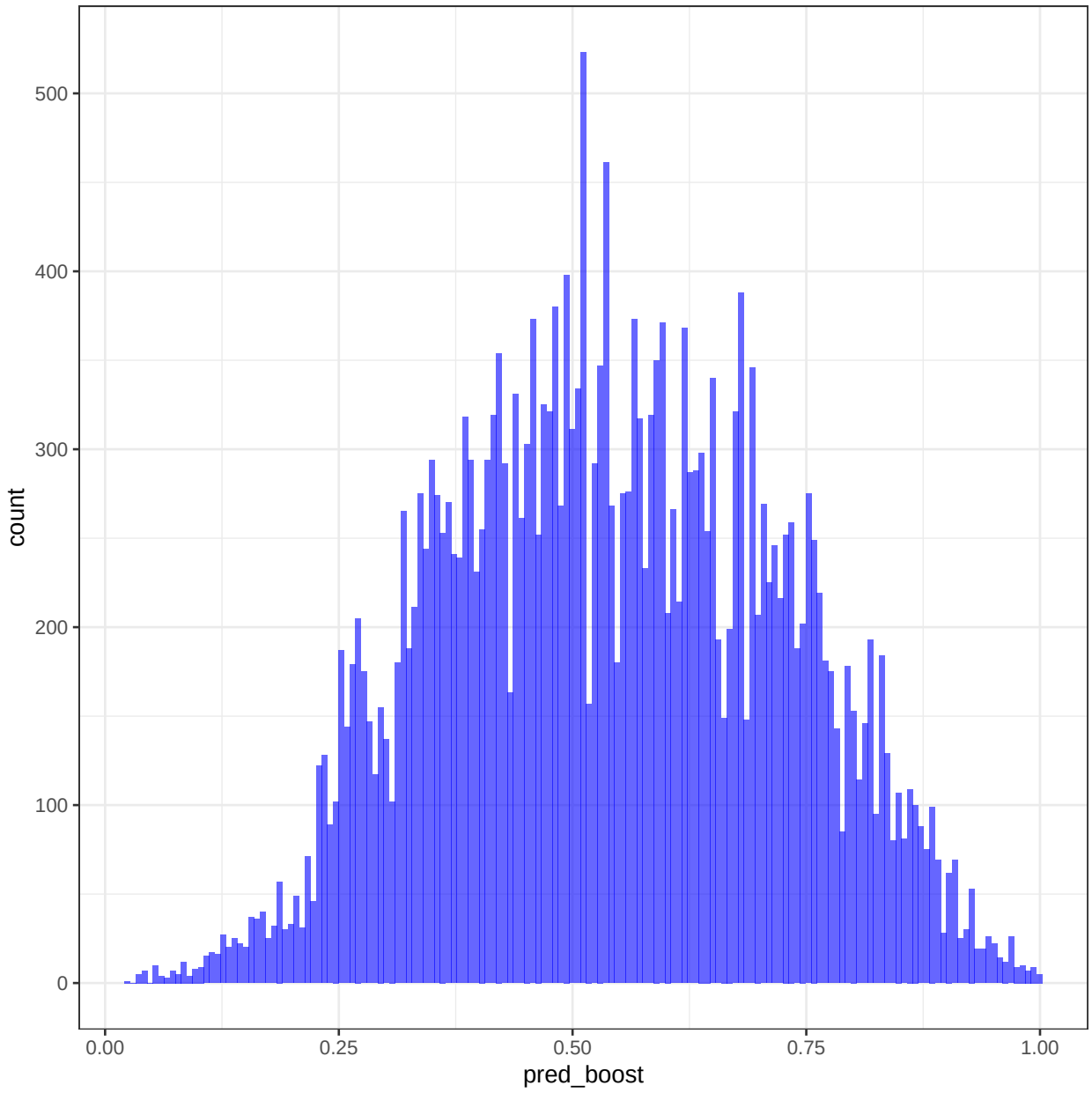

En este gráfico puede verse como se aumentaron los ponderadores, algunos de ellos de forma considerable. Si bien esto puede sonar alarmante los encuestados respondentes tienen que brindar información sobre los que no respondieron. Al tener tasas de respuesta bajas, la variación de los ponderadores, tiene que reflejar esa situación.

Algo a considerar, es que la estimación de la población obtenida con este nuevo sistema de ponderadores (3.415.329) siendo la original de (3.518.412) son similares entre sí, algo que no ocurría utilizando la estrategia de la parte 2, esta estimación crecia a casí el doble 6.582.193.

### Parte c

Con el mismo modelo de la parte anterior se ajustarán las no respuesta mediante propensiones estratificadas. Para ello se dividirán las propensiones en sus quintiles y luego a cada grupo se le asigna la propensión mediana.

Primero realizaremos un histograma de estas propensiones :




Podemos ver una distribución simétrica, centrada en valores poco mas grandes que $1/2$ aproximándose al valor de la tasa de respuesta.

A continuación calcularemos las propensiones estratificadas utilizando los quintiles de la distribución para agrupar y el estadístico utilizado para resumir el score será la mediana:

```
muestra %<>%
    mutate.(
        boost_class = cut(
            pred_boost,
            breaks = quantile(
                pred_boost,
                probs = seq(0,1,1/5)
            ),
            include.lowest = TRUE
        )
    ) %>%
    mutate.(
        ajuste_boost_clases = 1/median(pred_boost),
        .by = boost_class
    ) %>%
    mutate.(
        w_nr_boost_clases = R * w0 * ajuste_boost_clases
    )
```

```
muestra %>%
    as_survey_design(
        ids = id_hogar,
        weight = w_nr_boost_clases,
        strata = estrato
    ) %T>%
    assign(
        "diseño_nr_boost_clases",
        .,
        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_boost_clases,
      type = "kish"
    )
) %>%
assign(
  "est_ponderados_nr_clases",
  .,
  envir = .GlobalEnv
)
```

**Cuadro 7:** Estimaciones poblacionales usando ponderadores por no respuesta utilizando las propensiones del punto anterior por clases

| Variable | Estimación puntual | Error estandar | CV | deff | Efecto diseño de Kish |
|---|---|---|---|---|---|
| pobre | 0.089 | 0.004 | 0.047 | 3.171 | 1.153 |
| td | 0.083 | 0.004 | 0.044 | 1.257 | 1.153 |
| yprom | 21878.361 | 268.478 | 0.012 | 1.039 | 1.153 |

## Parte 3

Una vez realizado el ajuste por no respuesta, vamos a calibrar estos ponderadores en base conteos poblacionales por departamento, sexo y edad. Primero construiremos los totales marginales:

```
read_excel(
    here(
        "data",
        "dpto.xlsx"
    )
) %>%
rename.(
    personas_dpto = personas
) %>%
pull.(
    "personas_dpto"
) %>%
unique() %>%
assign(
    "total_dpto",
    .,
    envir = .GlobalEnv
)

edad_sexo <- read_excel(
    here(
        "data",
        "sexo_edad.xlsx"
    )
)

edad_sexo  %>%
    mutate.(
        total = hombres + mujeres,
        .keep = "unused"
    ) %>%
    mutate.(
        edad_tramo = cut(
            edad,
            breaks = c(0,14,20,25,30,40,50,60,Inf),
            right=FALSE
        )
    ) %>%
    summarize.(
        total = sum(total),
        .by = "edad_tramo"
    ) %>%
    rename.(
        total_edad = total
    ) %>%
    pull.(
```

```
        "total_edad"
    ) %>%
    unique() %>%
    assign(
        "total_edad",
        .,
        envir = .GlobalEnv
    )

edad_sexo  %>%
    summarize.(
        hombres = sum(hombres),
        mujeres = sum(mujeres)
    ) %>%
    pivot_longer.(
        names_to = "sexo",
        values_to = "valor"
    ) %>%
    rename.(
        total_sexo = valor
    ) %>%
    mutate.(
        sexo = ifelse.(
            sexo == "hombres",
            1,
            2
        )
    ) %>%
    pull.(
        "total_sexo"
    ) %>%
    unique() %>%
    assign(
        "total_sexo",
         .,
        envir = .GlobalEnv
    )



conteos <- c(
    sum(muestra$w0),
    total_dpto[-1],
    total_edad[-1],
    total_sexo[-1]
)


survey::calibrate(
    design = diseño_nr_boost_clases,
    formula = ~ as.factor(dpto) + edad_tramo + as.factor(sexo),
```

```
    population = conteos,
    calfun="raking",
    bounds
) -> r1

## Error in calibrate.survey.design2(design = diseño_nr_boost_clases, formula = ~as.factor(dpto
+ : objeto 'bounds' no encontrado

summary(weights(r1))

## Error in weights(r1): objeto 'r1' no encontrado
```

# Parte 4

## Referencias