

UNIVERSIDAD CATÓLICA DEL MAULE
FACULTAD DE CIENCIAS BÁSICAS
Ingeniería en Estadística



Evaluación 2

Integrantes: Isidora Allende Moraga - Francisca Troncoso Chacón

Asignatura: Manejo de software

Profesor: Pablo Jimenez Rodríguez

Fecha de entrega: November 30, 2025

Contents

1	Introducción	2
2	Ejercicio 3: Indicadores	2
2.1	Personas y hogares por año	2
2.2	ngresos del trabajo principal	2
3	Ejercicio 4: Comparación de eficiencia	3
3.1	¿Cuál estrategia es más eficiente y por qué?	3
3.2	¿Existen diferencias relevantes entre dplyr y data.table?	3
3.3	¿Usar map o apilar tablas genera diferencias notorias?	4

1 Introducción

El presente informe aborda el desarrollo de la Evaluación Unidad 3, utilizando las bases de datos de la Encuesta Suplementaria de Ingresos (ESI) de los años 2018 a 2022.

El objetivo principal es:

- Procesar y analizar las bases de datos entregadas por el INE.
- Generar indicadores descriptivos.
- Comparar estrategias de programación en términos de eficiencia.
- Presentar los resultados de forma clara y ordenada.

2 Ejercicio 3: Indicadores

2.1 Personas y hogares por año

A partir de cada base de datos de la ESI, se calcularon:

- Número de personas únicas (`idrph`).
- Número de hogares (`id_identificacion`).

Table 1: Número de personas y hogares por versión de la ESI

version	n_personas	n_hogares
esi-2018-personas	103741	34772
esi-2019-personas	96240	32664
esi-2020-personas	71935	24577
esi_2021	100433	35301
esi_2022	93103	33404

2.2 Ingresos del trabajo principal

Para quienes tienen ocupación de referencia (`ocup_ref == 1`), se resumen los ingresos:

- Mínimo
- Máximo
- Media
- Percentiles 10 y 90

Table 2: Estadísticos de ingresos del trabajo principal por año

version	min	max	media	p10	p90
esi-2018-personas	0	25200000	534929.4	100239.3	1003609
esi-2019-personas	0	20163695	544773.4	100058.1	1046496
esi-2020-personas	0	18045762	567410.5	100000.0	1177962
esi_2021	0	38206253	586360.4	140000.0	1114706
esi_2022	0	50000000	650711.3	148554.2	1237952

3 Ejercicio 4: Comparación de eficiencia

Se evaluaron 4 estrategias:

1. *dplyr_lista*
2. *dplyr_apilado*
3. *dt_lista*
4. *dt_apilado*

La comparación se realizó usando microbenchmark.

Table 3: Resumen del microbenchmark por estrategia (ms)

expr	min	lq	mean	median	uq	max	neval
dplyr_lista	206.180	218.698	440.591	265.353	569.800	942.922	5
dplyr_apilado	1402.008	1592.524	1813.555	1866.546	1919.720	2286.975	5
dt_lista	702.296	766.285	1124.784	857.228	945.951	2352.160	5
dt_apilado	1020.249	1104.657	1343.758	1134.814	1702.057	1757.012	5

3.1 ¿Cuál estrategia es más eficiente y por qué?

La estrategia que presentó el menor tiempo de ejecución, según la mediana reportada por el microbenchmark, fue:

- Estrategia: *r fast_expr*
- Tiempo mediano: *r round(fast_med, 3) ms*

Esta estrategia constituye la alternativa más eficiente para el cálculo de los indicadores, por lo que es la más recomendable en contextos donde se requiere rapidez, repetitividad o procesamiento de grandes volúmenes de datos.

3.2 ¿Existen diferencias relevantes entre dplyr y data.table?

La estrategia que exhibió el mayor tiempo de ejecución fue:

- Estrategia: *r slow_expr*

- Tiempo mediano: $r\ round(slow_med, 3)\ ms$

Debido a su mayor tiempo de procesamiento, esta estrategia no resulta adecuada cuando se busca optimizar el rendimiento o reducir tiempos de cómputo.

3.3 ¿Usar map o apilar tablas genera diferencias notorias?

Los resultados muestran diferencias claras entre las estrategias evaluadas. En términos generales:

- Las implementaciones basadas en `data.table` tienden a superar en rendimiento a aquellas que utilizan `dplyr`, lo cual coincide con la literatura y la optimización interna de cada paquete.
- Las estrategias que operan sobre tablas apiladas suelen mostrar mejor desempeño respecto de las que utilizan listas de tablas, dado que la operación sobre estructuras consolidadas reduce la sobrecarga de iteración.
- La estrategia identificada como más eficiente ($r\ fast_expr$) constituye la opción óptima para procesos de análisis sistemático o producción.

Entre las cuatro alternativas propuestas, la estrategia $r\ fast_expr$ es la más adecuada para trabajar con bases ESI en escenarios donde el rendimiento computacional sea un factor relevante.