Resultados Evaluación Unidad 3

1. Tabla de personas y hogares

version	num_personas	num_hogares
esi_2018	103741	34772
esi_2019	96240	32664
esi_2020	71935	24577
esi_2021	100433	35301
esi_2022	93103	33404

2. Estadísticos de ingresos

versión	mínimo	máximo	media	p10	p90
esi_2018	0	25.200.000	534.929	100.239	1.003.609
esi_2019	0	20.163.694	544.773	100.058	1.046.495
esi_2020	0	18.045.761	567.410	100.000	1.177.962
esi_2021	0	38.206.252	586.360	140.000	1.114.706
esi_2022	0	50.000.000	650.711	148.554	1.237.951

3. Eficiencia benchmark

expr	time
E1_Lista_Purrr_Dplyr	217115700
E3_Lista_Purrr_Data.table	276167702
E4_Apiladas_Data.table	775147202
E2_Apiladas_Dplyr	547911702
E4_Apiladas_Data.table	595868101
E4_Apiladas_Data.table	787219001
E1_Lista_Purrr_Dplyr	261813300
E2_Apiladas_Dplyr	229131702
E3_Lista_Purrr_Data.table	442833601
E3_Lista_Purrr_Data.table	651124701
E3_Lista_Purrr_Data.table	904905601
E2_Apiladas_Dplyr	398078801
E3_Lista_Purrr_Data.table	299895901
E1_Lista_Purrr_Dplyr	284103100
E4_Apiladas_Data.table	645767802
E1_Lista_Purrr_Dplyr	235526201
E2_Apiladas_Dplyr	334164201
E2_Apiladas_Dplyr	337515700
E4_Apiladas_Data.table	670417600
E1_Lista_Purrr_Dplyr	303038401

Etiquetas de fila	Promedio de time
E1_Lista_Purrr_Dplyr	260319340,4
E2_Apiladas_Dplyr	369360421,2
E3_Lista_Purrr_Data.table	514985501,2
E4_Apiladas_Data.table	694883941,2
Total general	**459887301**

Respuestas

1. ¿Cuál estrategia es más eficiente y por qué?

Según los resultados obtenidos en mi código, la estrategia más eficiente fue la Estrategia 1 (lista de tablas + purrr). Creemos que esto se debe a que, para el volumen de datos procesado, la librería dplyr gestionó las operaciones de forma optimizada sin incurrir en el costo computacional que implica la conversión de objetos a formato data.table en cada iteración, lo cual terminó afectando el rendimiento de las estrategias E3 y E4 en este ejercicio.

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

Sí, se observaron diferencias de tiempo muy relevantes. En este experimento particular, las implementaciones con dplyr resultaron ser más rápidas que las de data.table (casi la mitad del tiempo en algunos casos). Aunque teóricamente data.table suele ser superior en millones de filas, como vimos en clases, nuestros resultados indican que para estas bases de datos específicas dplyr tuvo un mejor rendimiento.

3. ¿Usar map o apilar tablas genera diferencias notorias?

Sí, genera diferencias notorias. Al comparar las estrategias, se observa que mantener las tablas en una lista y procesarlas con map (E1 y E3) fue más rápido que apilar todas las tablas en un solo data frame gigante (E2 y E4). Esto muestra que el costo de procesar una tabla unificada muy grande (apilada) fue mayor que el costo computacional de iterar sobre varios archivos más pequeños por separado.