Universidad de la República

Facultad de Ciencias Económicas y de Administración

Trabajo final de grado

---

metaSurvey

Paquete de R para el procesamiento de encuestas por muestreo con generación de recetas mediante metaprogramación y estimación de varianzas.

Estudiante:
Mauro Loprete

Tutora:
Dra. Natalia da Silva

Trabajo final de grado presentado como requisito para la obtención del titulo Licenciado en Estadística

Resumen

metaSurvey

por Mauro Loprete

El trabajo presenta metaSurvey, un paquete de R diseñado para mejorar el procesamiento de encuestas por muestreo y la estimación de parámetros poblacionales. Utiliza metaprogramación y técnicas de remuestreo para ofrecer resultados precisos, evaluar la incertidumbre y fomentar la reproducibilidad. A diferencia de otras bibliotecas, metaSurvey combina flexibilidad mediante metaprogramación con las capacidades de procesamiento de encuestas del paquete survey. Los objetivos incluyen proporcionar una herramienta útil, incorporar técnicas de remuestreo para usuarios no expertos, permitir la generación de 'recetas' personalizadas, y fomentar la contribución de la comunidad. Se destaca como alternativa a paquetes propietarios, enfocándose en la transparencia y reproducibilidad para mejorar la confiabilidad de las estimaciones poblacionales.

Agradecimientos

Acá le voy a agradcer a alguien?

Índice

Figuras

Tablas

Acá va la dedicatoria

Chapter 1

Introducción

En este documento se presenta el desarollo del paquete metaSurvey disponible en R (R Core Team, 2023). El objetivo principal de metaSurvey es permitir que el usuario pueda construir indicadores de manera reproducible y transparente, teniendo el usuario un control total sobre el proceso de transformación de los microdatos a indicadores.

A lo largo del documento se mencionan diferentes conceptos para el desarrollo del paquete, como la meta-programación, conceptos de Inferencia en Poblaciones finitas, esquema de trabajo para desarrollar paquetes en R, etc. Se presentarán ejemplos de cómo utilizar el paquete metaSurvey para construir indicadores de mercado de trabajo a partir de los microdatos de la Encuesta Continua de Hogares (**ECH**) del Instituto Nacional de Estadística de Uruguay (INE) y para mostrar su flexibilidad un ejemplo con la Encuesta Permanente de Hogares (**EPH**) del Instituto Nacional de Estadística y Censos de Argentina (INDEC).

La motivación principal del desarrollo del paquete fue la necesidad de contar con un paquete que permita al usuario tener un control total y transparente sobre el proceso de transformación de los microdatos a indicadores. En general, los paquetes que existen en R para el análisis de encuestas por muestreo son muy sensibles a la estructura y las variables que componen a la encuesta. En general, un cambio en la estructura de la encuesta implica una nueva versión del paquete utilizado para obtener los indicadores, siendo poco flexible a cambios en la estructura de la encuesta que en la practica pueden ser muy cambiantes y generalmente el usuario cuenta con una función de alto nivel que actua como caja negra donde no se permite modificar el código para adaptarlo a sus necesidades o saber cada paso que se realiza para obtener el indicador sin tener que leer el código fuente o la documentación adjunta.

El problema de sensibilidad a la estructura de la encuesta puede verse en el paquete *ech* (Detomasi, 2020) donde en el existen funciones para crear variables de mercado de trabajo, educación o ingresos pero estas funciones dependen de la existencia de ciertas variables en la encuesta donde la estructura puede cambiar de una versión a otra de la encuesta y sin revisar el cuerpo de la función no se sabe el proceso de construcción de variables. Algo similar ocurre con el paquete *eph* (Kozlowski et al., 2020), donde se tienen funciones de alto nivel que no permiten modificar el código para adaptarlo a sus necesidades o saber cada paso que se realiza para obtener el indicador sin tener inspeccionar a fondo el como se construyen las funciones del paquete.

Esta inspección del código fuente como puede ser consultar el repositorio de github del paquete o revisar la definición de la función, puede ser una tarea tediosa y que no garantiza que el usuario pueda entender el proceso de construcción de las variables ya

que puede ser que el código sea muy extenso o que el usuario no tenga el conocimiento suficiente para entender el código o se empleen ciertos frameworks que el usuario no conozca, como por ejemplo el uso de la librería dplyr (Wickham et al., 2023) o la librería tidyr (Wickham et al., 2024) que son librerías muy populares en R para el manejo de datos pero que el usuario puede no conocer o sea dificil aislar el proceso de construcción de variables y al funcionamiento especifico de la función donde puedan existir código para manejar la forma de presentación, esturcutura del objeto a devolver, etc. Un claro ejemplo de esto puede verse en el paquete tidycensus (Walker & Herman, 2024) donde existe una función para obtener datos sobre la migración de la comunidad Estado Unidense pero el usuario no puede y asilar el proceso de recodificación/construcción de variables sobre variables originales y la obtención de datos greográficos [1].

En este sentido, es importante que el usuario pueda tener un control total sobre el proceso de transformación de los microdatos a indicadores, ya que esto permite que el usuario pueda validar y entender el proceso de construcción de indicadores además de brindar una herramienta común libre de estilos de programación y definiendo con simples pasos el proceso de construcción de variables sintéticas, como puede ser recodificar variables creando grupos en base a criterios complejos, tratamiento de variables continuas como puede ser el ingreso salarial en base a metodología rigurosa y es crucial que sea transparente y entendible para el usuario.

En general obtener la información historica de indicadores es un proceso tedioso y que puede ser propenso a errores, en especial si provienen de encuentas donde su estructura y forma de preguntar o codificación puede cambiar en el tiempo siendo un proceso extenso y dificil de entender hasta llegar a la construcción de esta serie de indicadores y muchas veces diferentes usuarios hacen el mismo proceso de construcción de indicadores de manera independiente y sin compartir el código fuente o la metodología de construcción de indicadores ya que cada uno utiliza su propio estilo de programación o hasta diferentes paquetes estadísticos, en su mayoría propietarios como SPSS, SAS o STATA donde si bien el usuario puede compartir la sintaxis esta ligado a la sintaxis y depende de que el usuario tenga el software instalado y pueda correr el código.

Una vez claro el proceso de creación de variables también es importante tener en cuenta que al obtener indicadores se realiza un proceso de estimación de parámetros poblacionales y sus errores asociados, en este sentido es importante que el usuario no experto tenga de forma nativa una forma de obtener estimaciones puntuales y sus errores asociados de manera sencilla y brindar recomendaciones sobre la utilidad de la estimación en el caso de que se cuente con una variabilidad alta en la estimación ya que en general, obtener la estimación una vez culminado el proceso de preprocesamiento es relativamente sencillo pero puede ser que se reporte una estimación donde no exista un tamaño de muestra suficiente para obtener una estimación confiable y/o que la variabilidad de la estimación sea alta y no sea recomendable su uso.

En este sentido, el paquete metaSurvey pretende ser una herramienta relevante para el trabajo con encuestas buscando solucionar estas limitaciones ya que todo el proceso de transformación de los microdatos a indicadores se realiza a través de una

[1] Aquí puede verse el código fuente de la función `get_flow` del paquete tidycensus donde se puede en la linea 151 la recodificación de variables se hace con una tabla `mig_recodes` y al explorar el contenido puede verse como se recodifican las variables además de que la función también tiene código para manejar la forma de presentación de los datos, manipulación de datos geográficos y la estructura del objeto a devolver.

serie de funciones que permiten al usuario tener un control total y transparente sobre el proceso de transformación de los microdatos a indicadores. Además, metaSurvey permite que el usuario pueda realizar el proceso de transformación de los microdatos a indicadores de manera reproducible y transparente, ya que el usuario puede compartir el código y los datos utilizados para obtener los indicadores, mediante lo que denominamos steps y recipes, conformando así una especie de "recetario de cocina" para la construcción de indicadores, pudiendo compartir la construcción en forma visual como un DAG (Directed Acyclic Graph) que permite visualizar el proceso de construcción de indicadores sin tener que abrir un script de R. Además, metaSurvey permite que el usuario pueda obtener estimaciones puntuales y sus errores asociados de manera sencilla y brindar recomendaciones sobre la utilidad de la estimación en el caso de que se cuente con una variabilidad alta en la estimación.

El enfoque que permite la flexibilidad a la hora de construir los indicadores es la meta-programación. La meta-programación es un paradigma de programación que permite que un programa pueda modificar su estructura interna. En R, la meta-programación se realiza a través de las funciones eval y parse que permiten evaluar y parsear código de manera dinámica. En este sentido, metaSurvey utiliza la meta-programación para permitir que el usuario pueda modificar el código que se utiliza para transformar los microdatos a indicadores teniendo funciones similares a las que se utilizan en el paquete recipes de la librería tidymodels (Kuhn et al., 2024).

En los siguientes capitulos se mencionaran conceptos clave para el desarrollo del paquete, como la meta-programación, conceptos de Inferencia en Poblaciones finitas, esquema de trabajo para desarrollar paquetes en R, etc. A continuación se mencionarán diferentes antececentes y trabajos relacionados con el paquete metaSurvey donde se utiliza la meta-programación y herramientas en las que fue inspirado el paquete. Luego, se formalizaran diferentes conceptos sobre metodología para la estimación de parámetros poblaciones y su varianza y conceptos de meta-programación y como se utilizan en el desarollo del paquete. Para finalizar, se presentarán ejemplos de cómo utilizar el paquete metaSurvey para construir indicadores de mercado de trabajo a partir de los microdatos de la Encuesta Continua de Hogares (**ECH**) y para mostrar su flexibilidad un ejemplo con la Encuesta Permanente de Hogares (**EPH**).

Chapter 2

Marco conceptual

Chapter 3

Muestreo

Chapter 4

Metodología de varianzas

Chapter 5

Desarollo del paquete

Chapter 6

Infraestructura

- Infra
- Docker
- Kubernetes
- Tests
- Envío a CRAN

Chapter 7

Aplicación

- Mostrar Series Históricas
- Aplicación ANII
- Compartir recetas

Chapter 8

Conclusiones y trabajo a futuro

- Mostrar Series Históricas
- Aplicación ANII
- Compartir recetas

Soy un apaendice

Detomasi, G. M. &. R. (2020). *Ech: Caja de herramientas para procesar la encuesta continua de hogares*. https://github.com/calcita/ech

Kozlowski, D., Tiscornia, P., Weksler, G., Rosati, G., & Shokida, N. (2020). *Eph: Argentina's permanent household survey data and manipulation utilities*. https://holatam.github.io/eph/

Kuhn, M., Wickham, H., & Hvitfeldt, E. (2024). *Recipes: Preprocessing and feature engineering steps for modeling*. https://github.com/tidymodels/recipes

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Walker, K., & Herman, M. (2024). *Tidycensus: Load US census boundary and attribute data as 'tidyverse' and 'sf'-ready data frames*. https://walker-data.com/tidycensus/

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation*. https://dplyr.tidyverse.org

Wickham, H., Vaughan, D., & Girlich, M. (2024). *Tidyr: Tidy messy data*. https://tidyr.tidyverse.org