Propuesta de un procedimiento para la actualización de la tabla de cohortes de estudiantes del bachillerato de la Escuela de Estadística, UCR

Andrés E. Arguedas L.

B40535

Universidad de Costa Rica

Práctica Profesional II - 2017

Índice

[I. Introducción 2](#_Toc498690773)

[II. Principales aspectos metodológicos 4](#_Toc498690774)

[III. Resultados principales 7](#_Toc498690775)

[IV. Conclusiones 11](#_Toc498690776)

[Bibliografía 12](#_Toc498690777)

[Anexos 13](#_Toc498690778)

# Introducción

Tener información sobre la matrícula de estudiantes en la universidad, tanto en la carrera como en los cursos, brinda conocimiento valioso para la toma de decisiones en planificación a futuro de la carrera. Actualmente, la Oficina de Registro e Información (ORI) de la Universidad de Costa Rica es la entidad encargada de mantener los registros sobre los estudiantes matriculados en la carrera, la información sobre su récord académico y los cursos que ha llevado y el registro de las graduaciones. Por lo tanto, la propuesta para la creación de tablas de cohortes, con la ORI como fuente de información, surge dentro de la Escuela de Estadística como una iniciativa para tener información útil y de buena calidad para la toma de decisiones desde la dirección. Bajo esta misma idea, con base en las tablas de cohortes se puede planificar de mejor manera la cantidad y tamaño de grupos que se van a impartir para cada ciclo lectivo, además de indicadores valiosos como el porcentaje de estudiantes que se gradúa, la deserción de un año a otro, entre otros.

En el 2010, con seguimiento en el 2011, se inició un proyecto en el curso de Práctica Profesional I para analizar y crear las tablas de cohortes dentro de la Escuela. Para este análisis se usó el programa SPSS y la información era almacenada en un solo archivo, en el que cada fila representaba a un estudiante, donde las columnas contenían información personal del mismo y, posteriormente, información sobre todos los cursos que ha llevado. Esta forma de manejar la información, aunque dio resultados importantes, era complejo de manejar y actualizar. Por lo tanto, en el 2012, el profesor Daniel Antich propuso hacer la exportación de la base de datos a Access, para facilitar la actualización y manejo de los datos. Aun así, el sistema no era el adecuado, dado que él era la única persona que sabía cómo usar el sistema. Este problema se agravó a principios del 2017, cuando el profesor Antich se pensionó, por lo que no quedó dentro de la Escuela ningún respaldo ni información sobre las tablas de cohortes, que quedaron actualizadas al primer ciclo del 2016. Por lo tanto, surge la necesidad dentro de la Escuela de hacer un nuevo sistema para crear las tablas de cohorte, por lo que se inició en el segundo semestre de ese año un proyecto de práctica profesional para atender este problema.

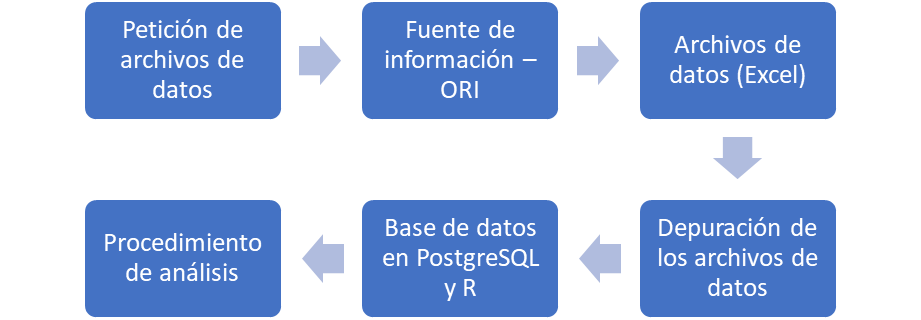
Este proyecto tiene un valioso aporte a la Escuela dado que las tablas de cohortes, además de ayudar en la toma de decisiones, brindan información valiosa para presentar evidencias durante el proceso de reacreditación ante SINAES. Adicionalmente, al hacer el nuevo sistema, una de las principales prioridades y preocupaciones es hacerlo de forma que sea fácil de usar y que tenga documentación para que, de esta forma, el sistema pueda ser usado por otra persona, tanto para actualizar los datos de las tablas como para generar estas.

Por lo tanto, el objetivo general de este proyecto es proponer un procedimiento computarizado para la actualización de la tabla de cohortes de estudiantes que ingresan al bachillerato en Estadística, que permita la fundamentación en la toma de decisiones estratégicas para efectos de matrícula y organización de los cursos de la carrera cada ciclo y año

Este informe está compuesto por una introducción, en la cual se detallan los antecedentes y justificación del proyecto, además de su objetivo y los riesgos; principales aspectos metodológicos, donde se especifican los detalles de los procedimientos y técnicas estadísticas usadas en el proyecto; resultados principales, donde se incluyen los productos finales del proyecto; conclusiones, donde se indica el impacto del proyecto, además del beneficio que este genera en la entidad; bibliografía y los anexos.

# Principales aspectos metodológicos

El primer paso del proyecto fue estudiar el funcionamiento del sistema anterior, desde la estructura de la base de datos, hasta la forma de construir la tabla, para construir un nuevo sistema basado en estas mismas ideas. Al haber revisado el sistema anterior, la primera decisión fue exportar la base de datos existente de Access a otro programa libre y que pudiera ser fácil conectarse con R, para hacer el posterior análisis de los datos. En este caso, se escogió el programa PostgreSQL, el cual es libre y hay paquetes dentro de R que permiten una fácil conexión a bases de datos en este programa. Por lo tanto, para este primer paso, lo más importante fue tener conocimiento sobre manejo de base de datos, tanto para entender la estructura y la funcionalidad de la primera base de datos, como para poder exportar los datos a otro programa. Este paso era necesario para el buen funcionamiento del sistema, tanto por la facilidad de hacer el análisis de los datos con R, como para tener más control sobre la base de datos y facilitar el acceso posterior a esta por parte de otras personas. Un esquema del proceso que usa el sistema de tablas de cohortes se presenta a continuación:



Al tener la base de datos lista, el siguiente paso es actualizar esta con los datos del 2014 en adelante. Dado que la información proviene de ORI y se tiene un archivo para cada año y cada ciclo lectivo, la estructura de la información no es necesariamente la misma, por lo que se tiene que hacer un proceso de depuración, limpieza y ordenamiento de la información para poderla agregar a la base de datos. Cabe resaltar que la base de datos está compuesta por tres tablas distintas: estudiantes, cursos y graduados. La tabla de estudiantes contiene la información personal de los estudiantes matriculados en la carrera, la tabla de cursos (notas) tiene los registros sobre los cursos que cada estudiante matriculado ha llevado y, por último, la tabla de graduados especifica la información sobre los graduados de la carrera. Estas tres tablas tienen variables e información distinta, por lo que el manejo de cada uno de estas y los procedimientos para agregar la información de los archivos de datos individuales varía entre cada una. Por lo tanto, en este paso, las principales habilidades y técnicas requeridas fueron la limpieza y manejo de archivos de datos. Es importante notar que este paso es el más importante del sistema, dado que, si se hace de una manera errónea, brinda una baja calidad de datos, por lo que los resultados también serán de baja calidad. Por lo anterior, y dada la distinta naturaleza de la información requerida por el sistema, es importante que este paso tenga una documentación adecuada y completa, para poder así facilitar el proceso de agregar datos posteriormente a la base de datos. Con respecto a lo anterior, es importante recordar que este sistema tiene como propósito ser actualizado cada vez que se tenga información nueva para actualizar las tablas de cohortes, por lo que este debe ser lo más general y adaptable posible, para no tener problemas de compatibilidad con la información a futuro. Teniendo la información de los archivos individuales ya procesada y ordenada, se procede a agregar a la base de datos, por lo que la información está actualizada al primer semestre del 2017.

Por último, teniendo la información actualizada en la base de datos, se procede a hacer el análisis de esta y crear las tablas de cohortes. Como se mencionó anteriormente, al tener la base de datos en PostgreSQL, se puede hacer una conexión a esta mediante R, por lo que todo el análisis de la información se realizó en R. La información se extrajo de la base de datos mediante una serie de consultas, desde R, aplicando filtros y clausulas lógicas, para luego ser agregado y analizado. El análisis se hizo mediante la agregación de los registros individuales, para cada año de ingreso a la carrera, para tener así una cohorte y poder hacer un seguimiento a lo largo del tiempo. La información para determinar el año de ingreso a la carrera se obtuvo tanto de la tabla de estudiantes como la de cursos, mientras que la información para el seguimiento de las cohortes proviene de las tablas de cursos y graduados. Adicionalmente, al haber registros de cursos que no están dentro del plan de estudios del Bachillerato en Estadística, se hizo un filtro a la información extraída de la base de datos para solo tomar en cuenta la información pertinente a la carrera y así tener información de mejor calidad. Dado lo anterior, en este paso, se necesitó tanto de manejo de base de datos, consultas a bases de datos y análisis y manejo de datos. Sin tener conocimiento sobre consultas a base de datos no se podría extraer la información necesaria de la base de datos y las técnicas de análisis de datos ayudan al manejo y arreglo de estos, después de ser extraídos, para poder brindar el producto final, que en este caso sería las tablas de cohortes. Por lo tanto, este paso es importante dado que amarra todos los pasos anteriores y brinda el resultado final, por lo que, de no hacerse de forma correcta, lo hecho anteriormente no serviría para nada.

# Resultados principales

Los dos resultados más importantes del proyecto son la propuesta del sistema para las tablas de cohortes (es decir, la base de datos y las funciones en R para el análisis de la información) y la creación del manual de uso del sistema. Con respecto a este primer resultado, se logró hacer la propuesta para un sistema para la elaboración de las tablas de cohortes, mediante el uso de la base de datos con la información y una serie de funciones y procedimientos en R para su análisis. Como se ha mencionado anteriormente, la base de datos se exportó desde Access a PostgreSQL, por lo que otro resultado de este proyecto es la base de datos, con la información actualizada y lista para ser analizada. Esta base de datos será manejada fuera de línea, en una computadora de la Escuela de Estadística, de tal forma que se asegure la confidencialidad de los datos y sea más fácil actualizarla y tener certeza sobre la veracidad de los datos.

Con respecto a los procedimientos y funciones para la depuración y análisis de los datos, estos se encuentran tanto en un repositorio de GitHub en línea (<https://github.com/andresarguedas/Practica-Estudiantes>) como en forma física, en caso de que estos sean requeridos por alguna persona para hacer el análisis y la creación de las tablas de cohortes. La función principal para el análisis de los datos (llamada **analisis**) toma como parámetro la conexión a una base de datos de PostgreSQL, el año de inicio y el año final del análisis deseado. La base de datos con la información de los estudiantes, da como resultado la tabla de cohortes ya hecha. Adicionalmente, la función permite exportar los resultados obtenidos de la función a un archivo de Excel y tiene otras opciones como la de poder escoger los años sobre los cuales hacer el análisis y obtener la tabla. El sistema creado permite el fácil análisis, almacenamiento y actualización de la información, además de dejar la puerta abierta a posibles mejoras a este sistema en el futuro. La Tabla 1 presenta el resultado final del sistema, después de hacer el análisis de los datos mediante las funciones propuestas.

**Tabla 1**

**Tabla de cohortes de la Escuela de Estadística, UCR, del 2000 al 2017.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Año lectivo | | | | | | | | | | | | | | | | | | Graduados |
| 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 |
| Año de ingreso | 2000 | 48 | 17 | 14 | 9 | 7 | 6 | 6 | 5 | 4 | 2 | 1 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 17 |
| 2001 |  | 20 | 11 | 8 | 5 | 4 | 4 | 3 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 9 |
| 2002 |  |  | 25 | 13 | 7 | 7 | 3 | 2 | 2 | 1 | 0 | 2 | 1 | 0 | 0 | 1 | 1 | 1 | 16 |
| 2003 |  |  |  | 25 | 17 | 12 | 10 | 9 | 5 | 3 | 2 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 12 |
| 2004 |  |  |  |  | 17 | 15 | 13 | 12 | 9 | 6 | 5 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 12 |
| 2005 |  |  |  |  |  | 26 | 12 | 7 | 6 | 2 | 3 | 4 | 1 | 1 | 0 | 2 | 0 | 0 | 9 |
| 2006 |  |  |  |  |  |  | 36 | 26 | 18 | 13 | 9 | 6 | 5 | 0 | 1 | 3 | 0 | 0 | 11 |
| 2007 |  |  |  |  |  |  |  | 38 | 24 | 15 | 17 | 11 | 11 | 10 | 8 | 9 | 6 | 4 | 12 |
| 2008 |  |  |  |  |  |  |  |  | 36 | 23 | 22 | 18 | 13 | 12 | 10 | 8 | 3 | 2 | 10 |
| 2009 |  |  |  |  |  |  |  |  |  | 46 | 33 | 29 | 30 | 20 | 14 | 10 | 8 | 6 | 14 |
| 2010 |  |  |  |  |  |  |  |  |  |  | 56 | 48 | 37 | 28 | 23 | 22 | 14 | 8 | 10 |
| 2011 |  |  |  |  |  |  |  |  |  |  |  | 64 | 49 | 39 | 38 | 31 | 23 | 18 | 10 |
| 2012 |  |  |  |  |  |  |  |  |  |  |  |  | 76 | 55 | 52 | 43 | 29 | 20 | 10 |
| 2013 |  |  |  |  |  |  |  |  |  |  |  |  |  | 93 | 70 | 55 | 47 | 29 | 5 |
| 2014 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 120 | 100 | 82 | 64 | 0 |
| 2015 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 85 | 70 | 57 | 0 |
| 2016 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 101 | 83 | 0 |
| 2017 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 103 | 0 |
| Retiro | | 0 | 31 | 12 | 20 | 19 | 9 | 22 | 20 | 33 | 39 | 18 | 25 | 37 | 59 | 46 | 51 | 87 | 91 |  |

Fuente: Sistema de tablas de cohorte de la Escuela de Estadística.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Año lectivo | | | | | | | | Graduados |
| 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 |
| Año de ingreso | 2010 | 56 | 48 | 37 | 28 | 23 | 22 | 14 | 8 | 10 |
| 2011 |  | 64 | 49 | 39 | 38 | 31 | 23 | 18 | 10 |
| 2012 |  |  | 76 | 55 | 52 | 43 | 29 | 20 | 10 |
| 2013 |  |  |  | 93 | 70 | 55 | 47 | 29 | 5 |
| 2014 |  |  |  |  | 120 | 100 | 82 | 64 | 0 |
| 2015 |  |  |  |  |  | 85 | 70 | 57 | 0 |
| 2016 |  |  |  |  |  |  | 101 | 83 | 0 |
| 2017 |  |  |  |  |  |  |  | 103 | 0 |
| Retiro | | 0 | 8 | 26 | 40 | 32 | 52 | 71 | 87 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Año lectivo | | | | | | | | Graduados |  |  |  |  |  |  |  |  |  |
| 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 |  |  |  |  |  |  |  |  |  |
| Año de ingreso | 2010 | 56 | 48 | 37 | 28 | 23 | 22 | 14 | 8 | 10 |  |  |  |  |  |  |  |  |  |
| 2011 |  | 64 | 49 | 39 | 38 | 31 | 23 | 18 | 10 |  |  |  |  |  |  |  |  |  |
| 2012 |  |  | 76 | 55 | 52 | 43 | 29 | 20 | 10 |  |  |  |  |  |  |  |  |  |
| 2013 |  |  |  | 93 | 70 | 55 | 47 | 29 | 5 |  |  |  |  |  |  |  |  |  |
| 2014 |  |  |  |  | 120 | 100 | 82 | 64 | 0 |  |  |  |  |  |  |  |  |  |
| 2015 |  |  |  |  |  | 85 | 70 | 57 | 0 |  |  |  |  |  |  |  |  |  |
| 2016 |  |  |  |  |  |  | 101 | 83 | 0 |  |  |  |  |  |  |  |  |  |
| 2017 |  |  |  |  |  |  |  | 103 | 0 |  |  |  |  |  |  |  |  |  |
| Retiro | | 0 | 8 | 26 | 40 | 32 | 52 | 71 | 87 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

En términos del manual de uso del sistema, este detalla cada uno de los pasos necesarios para usar el sistema, desde la petición de los datos, hasta el análisis de estos y la creación de las tablas de cohortes. Para cada uno de los pasos del proceso se presenta tanto los procedimientos necesarios como algunos ejemplos y código útil para poder realizar las distintas labores. Como se ha mencionado anteriormente, una de las prioridades de este proyecto era la elaboración de este manual, para así permitir que el sistema fuera de más fácil uso y acceso por distintas personas y, de esta forma, no depender de una sola persona para el mantenimiento y uso del sistema. Además, el manual fue sometido a revisión por parte de algunas personas que lo podrían usar para buscar así que este sea lo más entendible posible por las personas que lo deben usar en un futuro. Esta prueba incluyó los datos del primer semestre del 2017, para que así se usaran datos verdaderos para verificar que los procedimientos que en el manual se detallaban servirían. El manual de uso del sistema se presenta en este documento como Anexo 1.

Por último, el proyecto se concluyó a tiempo con lo establecido en el cronograma de trabajo. Uno de los factores que ayudó a la realización a tiempo del proyecto es la utilización de otro programa para la base de datos (PostgreSQL), además de tener conocimiento sobre consultas a bases de datos. Sin conocer sobre consultas, el análisis de la información se volvería más difícil y lento, lo que hubiera provocado problemas en el desarrollo del proyecto, tanto para obtener resultados como para crear la documentación sobre este proceso. Otro factor que ayudó al cumplimiento del proyecto es que ya se tenía conocimiento del problema y de algunas formas de solucionar este, por lo que este proyecto se montó sobre una solución ya implementada y la mejoró, lo que provocó que la realización del proyecto se hiciera de forma más expedita. De haber tenido que construir el sistema desde cero, se hubiera tomado más tiempo, dado que la información se tendría que adaptar de nuevo al sistema nuevo y este no necesariamente brindaría los resultados deseados.

# Conclusiones

Para empezar, es importante resaltar que se logró cumplir el objetivo del proyecto, dado que se hizo una propuesta exitosa de un procedimiento para la creación y análisis de las tablas de cohortes de la Escuela de Estadística. Con respecto a esto, el beneficiado principal es la Dirección de la Escuela, dado que este sistema brinda información valiosa para la toma de decisiones con respecto a la matrícula y la apertura de cursos. Adicionalmente, al ser un procedimiento documentado y basado en software libre, es fácil de repetir y mantener actualizado, por lo que el procedimiento debería tener valor y uso en el futuro. En este caso, la segunda beneficiada sería la Escuela como tal, de forma específica, tanto en el proceso como en la Comisión de Acreditación, dado que el sistema es capaz de brindar algunas evidencias valiosas y necesarias durante el proceso de reacreditación.

Adicionalmente, se lograron hacer mejoras con respecto al sistema anterior, como la posibilidad de exportar los resultados en un archivo de Excel, poder escoger solamente ciertos años para hacer el análisis y el cálculo de ciertos indicadores importantes para la fundamentación de la toma de decisiones. Estas mejoras dan un valor agregado al sistema, ya que no solamente logra hacer lo que el sistema anterior hacía, sino que también tiene otras funcionalidades que no estaban presentes en el sistema anterior. Por lo tanto, este proyecto dio como resultado un sistema útil y novedoso, que llegó a mejorar el que se usaba anteriormente y brinda información necesaria para la toma de decisiones, de forma fácil y actualizada.

# Bibliografía

* Conway, J., Eddelbuettel, D., Kumar-Prayaga, S., Nishiyama, T. & Tiffin N. (2017). *RPostgreSQL: R Interface to the 'PostgreSQL' Database System*. Recuperado de <https://CRAN.R-project.org/package=RPostgreSQL>.
* Francois, R., Henry, L., Müller, K. & Wickham, H. (2017). *dplyr: A Grammar of Data Manipulation*. Recuperado de <https://CRAN.R-project.org/package=dplyr>.
* Grolemund, G. & Wickham, H. (2011). *Dates and Times Made Easy with lubridate*. Journal of Statistical Software. 40 (3), 1-25. Recuperado de <http://www.jstatsoft.org/v40/i03/>.
* Milton-Bache, S. & Wickham, H. (2014). *magrittr: A Forward-Pipe Operator for R.* Recuperado de <https://CRAN.R-project.org/package=magrittr>.
* Petchey, C. (2015, 16 de junio). *Reading data and code from an online github repository.* Recuperado de <https://github.com/opetchey/RREEBES/wiki/Reading-data-and-code-from-an-online-github-repository>.
* R Core Team (2017). *R: A language and environment for statistical computing*. (Versión 3.4.2). R Foundation for Statistical Computing, Vienna, Austria. Recuperado de <https://www.R-project.org/>.

# Anexos

**Anexo 1: Manual de uso del sistema de tablas de cohortes**

## Petición de los datos

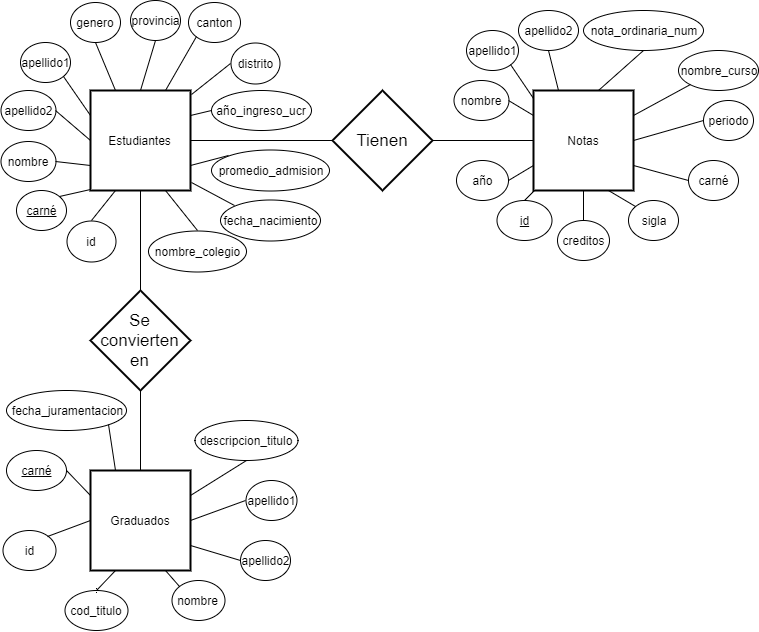
Al tratarse de registros administrativos sobre estudiantes, la fuente de la información es la Oficina de Registro e Información (ORI) de la Universidad de Costa Rica. Desde la dirección de la Escuela de Estadística, se envía una carta cada semestre para pedir los archivos con los datos necesarios para poder utilizar este sistema. Un ejemplo de esta carta se presenta al final de este manual como Figura 1. Como se explicará posteriormente, es importante hacer la petición de tres archivos distintos: la información personal de los matriculados en la carrera (referida como Estudiantes), las notas de cada estudiante matriculado en Estadística en cada curso (referida como Notas) y la información de los graduados de la carrera en ese período (referida como Graduados). Anteriormente, se trabajaba con archivos en los cuales cada fila representaba un estudiante y las columnas contenían su información personal e información sobre los cursos en los que estaba matriculado, pero al pasar el almacenamiento de la información a una base de datos, la estructura de los archivos y la forma en los que se trabaja con estos cambió. Estos archivos deben corresponder al año y ciclo lectivo solicitado y su solicitud se debe hacer lo más rápido posible, dado que ORI elimina periódicamente de sus registros a los estudiantes graduados, por lo que, de tener algún retraso en la solicitud, se puede perder información que no se puede recuperar. Adicionalmente, es importante que cada uno de los archivos contenga, por lo menos, las siguientes variables:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Estudiantes** | **Notas** | **Graduados** |
| Variables | Carné | Carné | Carné |
| Nombre | Nombre | Nombre |
| Apellidos | Apellidos | Apellidos |
| Género | Nota | Fecha de juramentación |
| Provincia | Nombre del curso | Descripción del título |
| Cantón | Periodo |  |
| Distrito | Sigla |  |
| Promedio de admisión | Créditos |  |
| Fecha de nacimiento | Año |  |
| Colegio de procedencia |  |  |

Al hacer la solicitud de los datos a ORI y habiéndolos recibido, se procede a la depuración de estos, dado que el formato en cual son enviados por ORI no son necesariamente los mismos siempre, por lo que puede haber problemas al momento de exportarlos a la base de datos. Algunos ejemplos de lo que se debe depurar son la obtención del año de ingreso mediante el número de carné, la eliminación de tildes y caracteres especiales, cambio de notas alfanuméricas a numéricas y la obtención de carnés faltantes. Por lo anteriormente mencionado, es importante comprender la estructura de la base de datos y los procesos necesarios para depurar los archivos de datos y, posteriormente, poder exportar estos a la base de datos.

## Estructura de la base de datos y depuración de los datos

Cuando se hizo la exportación original de los datos de las hojas en Excel a una base de datos, inicialmente se usó el programa Access. Posteriormente, al querer renovar el sistema y hacer parte del análisis de los datos en R, se dieron varios problemas con la conexión entre R y Access, lo que dificultaba de forma importante este análisis. Por lo anterior, se decidió exportar la base de datos desde Access hacia PostgreSQL, manteniendo la estructura de entidad-relación de la base de datos original, que se presenta en la siguiente figura:



El objetivo de la depuración de los datos es que se comparta una estructura común entre los distintos archivos, tanto para años como ciclos lectivos distintos. Como se mencionó anteriormente, al tener estos almacenados en una base de datos, es importante que los archivos individuales den, al menos, la información necesaria para alimentar la base de datos. Adicionalmente, algunas de las variables, como el año de ingreso a la universidad, deben ser construidas con base en otras variables, por lo que este procedimiento también se detalla aquí. El proceso de depuración aquí descrito se basa en el uso del software R, por lo que los ejemplos presentados corresponden a funciones y comandos de este lenguaje. Aun así, es importante mencionar que el proceso de depuración puede ser realizado en cualquier programa o lenguaje de programación, lo importante es que el producto final sea compatible con la información en la base de datos. Al tratarse de tres tablas distintas de la base de datos, se procede a explicar el procedimiento de depuración de cada una de estas por separado, aunque hay ciertas variables que se comparten entre las tablas.

### Estudiantes

* **ano\_ingreso\_ucr:** Variable de tipo carácter que contiene el año de ingreso a la UCR. En caso de no tener esta variable, se puede deducir del carné mediante los siguientes comandos y usando la librería **dplyr**:

library**(**dplyr**)**

substr**(**carne, start **=** 1, stop **=** 2**)** %>%

gsub**(**"A", "200", .**)** %>%

gsub**(**"B", "201", .**)**

En este caso, **carne** sería el número de carné y lo que devuelve este comando es el año de ingreso a la UCR. Esta variable no debería contener valores mayores al año actual y el valor 1900 se podría usar como un NA, aunque teóricamente todos los estudiantes tienen un carné, por lo que no debería haber valores faltantes en esta variable.

* **carne:** Variable de tipo carácter que contiene el número de carné de cada estudiante. Debe empezar con dos números, seguido por tres números, o con una A o B, seguida por cuatro números. El número de carné es único, por lo que no deben haber duplicados en esta variable si todas las personas son distintas. En caso de no contar con el número de carné, se puede hacer un cruce con los registros de las notas o los graduados para intentar solucionar este problema.
* **apellido1:** Variable de tipo carácter que contiene el primer apellido de la persona, por lo general en mayúsculas. Para evitar ciertos problemas que se pueden presentar con las tildes y la ñ, estas se deben reemplazar por otros caracteres, las letras sin tildes y la n, respectivamente. Lo anterior se puede hacer mediante los siguientes comandos:

toupper**(**apellido1**)** %>%

chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

Este comando devuelve el apellido, en mayúsculas, pero sin tildes ni la ñ, que es lo que se quería.

* **apellido2:** Variable de tipo carácter que contiene el segundo apellido de la persona, por lo general en mayúsculas. El análisis de esta variable es igual al de **apellido1**.
* **nombre:** Variable de tipo carácter que contiene el nombre de la persona, por lo general en mayúsculas. El análisis de esta variable es igual al de **apellido1**.
* **provincia:** Variable de tipo carácter que contiene la provincia de residencia de la persona, por lo general en mayúsculas. El análisis de esta variable es igual al de **apellido1**. Los valores faltantes se indican como **“PROVINCIA INVALIDA”.**
* **canton:** Variable de tipo carácter que contiene el cantón de residencia de la persona, por lo general en mayúsculas. El análisis de esta variable es igual al de **apellido1**. Los valores faltantes se indican como **“CANTON INVALIDO”.**
* **distrito:** Variable de tipo carácter que contiene el distrito de residencia de la persona, por lo general en mayúsculas. El análisis de esta variable es igual al de **apellido1**. Los valores faltantes se indican como **“DISTRITO INVALIDO”.**
* **genero:** Variable de tipo carácter que contiene el sexo de la persona, en mayúsculas. Todos los valores de esta variable deberían ser **“MASCULINO”** o **“FEMENINO”**.
* **fecha\_nacimiento:** Variable de tipo fecha, la cual contiene la fecha de nacimiento del estudiante, en formato AAAA-MM-DD. Por lo general, esta variable viene originalmente en formato DD/MM/AAAA y se lee como una variable de tipo carácter, por lo que el primer paso es convertirlo en formato de fecha, para luego cambiar la forma en la que se presenta la información. Lo anterior se puede lograr mediante los siguientes comandos usando la librería **lubridate**:

library**(**lubridate**)**

dmy**(**fecha\_nacimiento**)** %>%

ymd**()**

El resultado de este comando da un objeto con formato de fecha y con la forma AAAA-MM-DD, que es el resultado que se necesita para poder agregar esta información a la base de datos.

* **nombre\_colegio:** Variable de tipo carácter que contiene el nombre del colegio del cual se graduó el estudiante, por lo general en mayúsculas. El análisis de esta variable es igual al de **apellido1**.
* **promedio\_admision:** Variable de tipo numérica que contiene el promedio de admisión a la UCR de la persona. Este número debería ser positivo y menor o igual a 800, excepto en el caso de los NA, que deben ser codificados como 999.

### Notas

* **id:** Variable numérica secuencial, que empieza en 1 y es única para cada registro. Esta variable no se puede crear hasta que se hace el proceso de exportación de los datos, dado que no se conoce si hay registros duplicados en la información con respecto a lo que ya está en la base de datos.
* **carne:** Igual a la variable **carne** de la tabla de estudiantes.
* **sigla:** Variable de tipo carácter que contiene la sigla del curso correspondiente, por lo general en mayúsculas. Esta variable tiene una o dos letras al inicio, representando la escuela que lo imparte, seguido por cuatro números, para identificar el curso dentro de los impartidos por esa misma escuela. La sigla es una variable importante para el análisis de los datos, dado que con base en esta se selecciona los cursos que están dentro del plan de estudios del bachillerato, por lo que es importante que, no haya valores faltantes en esta variable. Por lo anterior, en caso de haber algún registro sin sigla, se puede hacer una comparación con base en el nombre del curso, para ver si hay algún registro de un curso que se llame igual y que si tenga sigla, para poder usar esta.
* **nombre\_curso:** Variable de tipo carácter que contiene el nombre del curso, por lo general en mayúsculas. Esta variable puede no ser muy confiable, dado que puede presentar problemas por cambio del nombre del curso, pero no de las siglas, o problemas por tildes o caracteres especiales, por lo que es preferible usar la variable **sigla** para el análisis de los datos, dado que esta es más confiable. En caso de no tener el nombre del curso en algún registro, se puede hacer una comparación con base en la sigla respectiva, para poder obtener así el nombre del curso.
* **nota\_ordinaria\_num:** Variable de tipo numérica que contiene la nota que la persona obtuvo en el curso. Esta variable tomar valores entre 0 y 10, siendo 5 el único decimal permitido. Normalmente esta variable viene en formato alfanumérico, es decir, contiene “AP” si el curso fue aprobado, “RM” si fue retirado, etc. Dado que se quiere transformar esta variable en una numérica, se toman todos los valores que no son numéricos como 0, aunque esto no es del todo la solución más apropiada. Los siguientes comandos, usando el operador **%<>%** de la librería **magrittr** son una solución a este problema:

library**(**magrittr**)**

nota\_ordinaria\_num %<>% as.numeric**()**

nota\_ordinaria\_num**[**is.na**(**nota\_ordinaria\_num**)]** **<-** 0

Como se puede notar, el comando anterior regresa las notas en formato numérico y, si la nota era de tipo carácter, esta se convierte en NA, por lo que los NA’s se definen posteriormente como 0. Teóricamente no debe haber valores faltantes en esta variable.

* **creditos:** Variable de tipo numérica que contiene la cantidad de créditos que vale el curso. Esta variable debe ser entera y, en caso de no tenerla, se puede hacer una comparación con base en el nombre y la sigla del curso para obtenerla.
* **periodo:** Variable de tipo carácter que contiene el ciclo lectivo en que se impartió en curso. Los únicos valores que esta variable debe tomar son **“1”**, **“2”** y **“3”**, correspondientes al primer ciclo lectivo, segundo ciclo lectivo y verano, respectivamente. En caso de tener valores faltantes en un archivo, esta variable se puede construir fácilmente si se conoce el ciclo lectivo al cual el archivo corresponde. Teóricamente no debe haber NAs en esta variable.
* **ano:** Variable de tipo numérica que contiene el año en que se impartió el curso. El valor mínimo de esta variable debe ser 2000 y no puede tener valores mayores al año actual. Al igual que en la variable **periodo**, esta variable es fácil de construir si se conoce el año del cual provienen los datos. Teóricamente no debe haber NAs en esta variable.

### Graduados

* **carne:** Igual a la variable **carne** de la tabla de estudiantes.
* **apellido1:** Igual a la variable **apellido1** de la tabla de estudiantes.
* **apellido2:** Igual a la variable **apellido2** de la tabla de estudiantes.
* **nombre:** Igual a la variable **nombre** de la tabla de estudiantes.
* **fecha\_juramentacion:** Variable de tipo fecha que contiene la fecha en la cual la persona se graduó. El análisis de esta variable es similar al de la variable **fecha\_nacimiento** de la tabla de estudiantes. Esta variable es importante para el análisis de los datos, por lo que no debería haber valores faltantes y, aunque sea, tener el año de graduación, dado que el mes y el día no son muy importantes.
* **cod\_titulo:** Variable de tipo carácter que contiene el código del título. No es una variable muy importante para el análisis de los datos.
* **descripcion\_titulo:** Variable de tipo carácter que contiene la descripción del título. No es una variable muy importante para el análisis de los datos.

### Ejemplo

Para ilustrar los procedimientos anteriormente descritos, se va a hacer un ejemplo de la depuración de datos ficticios. Estos datos se pueden obtener en el siguiente link (<https://github.com/andresarguedas/Practica-Estudiantes/tree/master/data>) y están guardados en tres archivos distintos, uno para estudiantes, otro para graduados y el último con las notas de los cursos. Para empezar, es necesario cargar algunas librerías que son necesarias para la depuración de los datos.

library**(**RCurl**)**

library**(**dplyr**)**

library**(**lubridate**)**

library**(**magrittr**)**

options**(**scipen **=** 999**)**

El paquete **RCurl** permite cargar los datos para los ejemplos directamente desde GitHub, **dplyr** da herramientas útiles para el manejo de datos, como lo son las funciones *select* y el operador *%>%*, **lubridate** brinda funciones para ayudar en el manejo de fechas y **magrittr** brinda el operador *%<>%*. Por último, el comando *options(scipen = 999)* permite que los números no se escriban en notación científica, evitando así problemas con números muy grandes. Teniendo estos paquetes cargados, podemos empezar la depuración de los datos.

#### Estudiantes

El primer paso es cargar los datos para la depuración.

est **<-** read.csv2**(**text **=** getURL**(**"https://raw.githubusercontent.com/andresarguedas/Practica-Estudiantes/master/data/estudiantes%20p.csv"**))**

En este caso, el objeto **est** contiene la información de los estudiantes. Se puede ver los nombres de las columnas de este objeto mediante el comando **names**.

names**(**est**)**

"X.U.FEFF.carne" "apellido1" "apellido2" "nombre" "genero" "provincia" "canton"

"distrito" "promedio" "nacimiento" "colegio"

Como se puede notar, el nombre de la primera columna está escrito de forma extraña y esto se debe a problemas que tiene R al cargar este tipo de archivos. Para arreglar esto, se puede especificar manualmente el nombre de esa columna, se muestra a continuación. Además, se le debe cambiar el nombre a la variable del promedio de admisión de **promedio** a **promedio\_admision**.

names**(**est**)[**1**]** **<-** "carne"

names**(**est**)[**9**]** **<-** "promedio\_admision"

Teniendo los nombres de las columnas bien, procedemos a la depuración de cada una de las variables del archivo. Para estos efectos, vamos a empezar creando la variable **ano\_ingreso\_ucr**.

est**$**ano\_ingreso\_ucr **<-** substr**(**est**$**carne, start **=** 1, stop **=** 2**)** %>%

gsub**(**"A", "200", .**)** %>%

gsub**(**"B", "201", .**)**

Es importante notar que el año de ingreso de un estudiante a la UCR se puede extraer de su número carné. Este puede venir de dos formas distintas, empezar con un número o empezar con la letra A o B. En el primer caso, el año de ingreso se puede obtener al agregar un 19 antes del carné y viendo los primeros cuatro dígitos, es decir, un año del siglo XX que representa el año de ingreso a la UCR. En el segundo caso, la A representa que la persona entró en algún año entre el 2000 y el 2009 y el primer digito indica cual año de estos es, mientras que, para el caso de los carnés que empiezan con B, estos son personas que ingresaron entre el 2010 y el 2019. Por lo tanto, para extraer el año de ingreso de un carné de ese tipo, es cuestión de agregar un 200 o un 201 en lugar de la letra correspondiente y ver los primeros cuatro dígitos, teniendo así un año entre el 2000 y el 2019, correspondiente al año de ingreso del estudiante. Por lo tanto, el comando anterior extrae el año de ingreso del número de carné, seleccionando los primeros dos caracteres del carné (los que especifican el año de ingreso) y sustituye la A por un 200 y la B por un 201, de tal forma que el resultado final es el año de ingreso a la universidad. Procedemos con la variable **carne**.

est**$**carne

A9999 B5000 B4999 B1999 B2000 A3000 B0000 A9000 A5999 B3000

Levels**:** A3000 A5999 A9000 A9999 B0000 B1999 B2000 B3000 B4999 B5000

Parece que la variable **carne** está bien y no tiene ningún problema, por lo que se puede seguir con la depuración de las demás variables. A continuación, se presenta la forma de depurar las variables **apellido1**, **apellido2**, **nombre**, **provincia**, **canton**, **distrito** y **nombre\_colegio**. Como se mencionó en la descripción de las variables de la sección 2.1, la forma de depurar estas variables es la misma, dado que primero se ponen en mayúscula y, posteriormente, se eliminan los caracteres especiales y las tildes. Este primer procedimiento se hace mediante la función **toupper** y el segundo se hace al sustituir las letras con tildes y la eñe por la letra equivalente sin tilde y una n, lo cual se hace mediante el comando **chartr(“ÁÉÍÓÚÑ”, “AEIOUN”)**. Cada variable se presenta en su forma original y después de aplicar los comandos, para poder ver la diferencia entre ambos.

est**$**apellido1

Pérez Grillo García López Bermúdez Rodríguez Fernández Cruz Álvarez Gamboa

Levels**:** Álvarez Bermúdez Cruz Fernández Gamboa García Grillo López Pérez Rodríguez

est**$**apellido1 %<>% toupper**()** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

est**$**apellido1

"PEREZ" "GRILLO" "GARCIA" "LOPEZ" "BERMUDEZ" "RODRIGUEZ" "FERNANDEZ" "CRUZ" "ALVAREZ" "GAMBOA"

est**$**apellido2

Pérez López García Jirón Saenz Hernández Salas Cruz Jiménez Zeledón

Levels**:** Cruz García Hernández Jiménez Jirón López Pérez Saenz Salas Zeledón

est**$**apellido2 %<>% toupper**()** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

est**$**apellido2

"PEREZ" "LOPEZ" "GARCIA" "JIRON" "SAENZ" "HERNANDEZ" "SALAS" "CRUZ" "JIMENEZ" "ZELEDON"

est**$**nombre

Juan Pepe Juana Daniel Laura Pablo Yolanda Penelope Paola María

Levels**:** Daniel Juan Juana Laura María Pablo Paola Penelope Pepe Yolanda

est**$**nombre %<>% toupper**()** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

est**$**nombre

"JUAN" "PEPE" "JUANA" "DANIEL" "LAURA" "PABLO" "YOLANDA" "PENELOPE" "PAOLA" "MARIA"

est**$**provincia

San José San José Limón San José Heredia Alajuela San José Puntarenas San José Cartago

Levels**:** Alajuela Cartago Heredia Limón Puntarenas San José

est**$**provincia %<>% toupper**()** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

est**$**provincia

"SAN JOSE" "SAN JOSE" "LIMON" "SAN JOSE" "HEREDIA" "ALAJUELA" "SAN JOSE" "PUNTARENAS" "SAN JOSE"

"CARTAGO"

est**$**canton

Desamparados Central Guácimo Central San Pablo Central Santa Ana Osa Moravia

Turrialba

Levels**:** Central Desamparados Guácimo Moravia Osa San Pablo Santa Ana Turrialba

est**$**canton %<>% toupper**()** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

est**$**canton

"DESAMPARADOS" "CENTRAL" "GUACIMO" "CENTRAL" "SAN PABLO" "CENTRAL" "SANTA ANA" "OSA"

"MORAVIA" "TURRIALBA"

est**$**distrito

Desamparados Hatillo Guácimo Zapote San Pablo Alajuela Piedades Palmar San Jerónimo

Turrialba

Levels**:** Alajuela Desamparados Guácimo Hatillo Palmar Piedades San Jerónimo San Pablo Turrialba Zapote

est**$**distrito %<>% toupper**()** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

est**$**distrito

"DESAMPARADOS" "HATILLO" "GUACIMO" "ZAPOTE" "SAN PABLO" "ALAJUELA" "PIEDADES" "PALMAR"

"SAN JERONIMO" "TURRIALBA"

est**$**colegio

Colegio los Patitos Colegio los Patitos Colegio los Patitos Colegio los Patitos Colegio los Patitos Colegio los Patitos

Colegio los Patitos Colegio los Patitos Colegio los Patitos Colegio los Patitos

Levels**:** Colegio los Patitos

est**$**nombre\_colegio **<-** toupper**(**est**$**colegio**)** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

est**$**nombre\_colegio

"COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS"

"COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS" "COLEGIO LOS PATITOS"

Habiendo depurado las variables anteriores, solo falta depurar las variables **genero** y **fecha\_nacimiento**. Para esta primera interesa que solamente haya dos opciones, *MASCULINO*y *FEMENINO*.

est**$**genero

MASCULINO MASCULINO FEMENINO MASCULINO FEMENINO MASCULINO FEMENINO FEMENINO FEMENINO FEMENINO

Levels**:** FEMENINO MASCULINO

Como se puede notar, parece que la variable **genero** no tiene problemas, por lo que se puede proceder a depurar la variable **fecha\_nacimiento**.

est**$**nacimiento

29**/**08**/**87 11**/**07**/**93 20**/**07**/**94 10**/**05**/**95 26**/**07**/**89 04**/**03**/**96 03**/**02**/**99 01**/**12**/**92 02**/**06**/**74 27**/**01**/**97

Levels**:** 01**/**12**/**92 02**/**06**/**74 03**/**02**/**99 04**/**03**/**96 10**/**05**/**95 11**/**07**/**93 20**/**07**/**94 26**/**07**/**89 27**/**01**/**97 29**/**08**/**87

est**$**fecha\_nacimiento **<-** ymd**(**dmy**(**est**$**nacimiento**))**

est**$**fecha\_nacimiento

"1987-08-29" "1993-07-11" "1994-07-20" "1995-05-10" "1989-07-26" "1996-03-04" "1999-02-03" "1992-12-01" "1974-06-02"

"1997-01-27"

Los comandos anteriores, mediante el paquete **lubridate**, transforman el vector de caracteres iniciales en un objeto de tipo fecha, el cual es el que se necesita en la base de datos. Es importante notar que originalmente los datos tenían formato DD/MM/AA, por lo que se usó el comando **dmy** y luego, para tenerlos en formato AAAA-MM-DD, se usó el comando **ymd**. Por lo tanto, habiendo depurado esta variable, ya tenemos la información de los estudiantes depurada y lista para ser exportada a la base de datos.

#### Notas

Al igual que en el ejemplo de estudiantes, el primer paso es cargar los datos para poder realizar el ejemplo.

notas **<-**read.csv2**(**text **=** getURL**(**"https://raw.githubusercontent.com/andresarguedas/Practica-Estudiantes/master/data/notas%20p.csv"**))**

Teniendo los datos cargados, revisamos los nombres de las columnas para corroborar que se tiene la información necesaria y que no hayan errores.

names**(**notas**)**

"X.U.FEFF.carne" "apellido1" "apellido2" "nombre" "sigla" "nombre\_curso" "nota"

"creditos"

names**(**notas**)[**1**]** **<-** "carne"

Al igual que en ejemplo con la información de estudiantes, el título de la columna con los números de carné tiene problemas, por lo que se puede arreglar manualmente. Ahora, la siguiente variable a depurar es **nombre\_curso** la cual, como se mencionó en la sección 2.2, se debe poner en mayúsculas y eliminar los caracteres especiales. Como se explicó en el ejemplo de estudiantes, esto se puede hacer mediante los comandos **toupper** y **chartr(“ÁÉÍÓÚÑ”, “AEIOUN”).**

notas**$**nombre\_curso

MÉTODOS ESTADÍSTICOS

ESTADÍSTICA COMPUTACIONAL I

MODELOS PROBABILÍSTICOS DISCRETOS

PRINCIPIOS DE FINANZAS

ENTORNO LEGAL DE LAS EMPRESAS I

ESTRATEGIAS DE LECTURA EN INGLÉS I **(**PARA OTRAS CARRERAS**)**

ÁLGEBRA LINEAL

MÉTODOS ESTADÍSTICOS

ESTADÍSTICA COMPUTACIONAL I

CURSO INTEGRADO DE HUMANIDADES II

CÁLCULO PARA CIENCIAS ECONÓMICAS I

MÉTODOS ESTADÍSTICOS

ÁLGEBRA LINEAL

SEM. DE REAL. NAC. 2**:** SALUD COMUNITARIA

LITERATURA ESPAÑOLA III

PSICOLOGÍA GENERAL I

SEMINARIO DE REALIDAD NACIONAL I DERECHOS HUMANOS Y SEGURIDAD CIUDADANA

CÁLCULO PARA CIENCIAS ECONÓMICAS I

SEM. REAL. NAC.1**:** DESARROLLO COMUNAL

INTRODUCCIÓN A LA ECONOMÍA

ESTADÍSTICA INTRODUCTORIA I

ITALIANO BÁSICO I

INTRODUCCIÓN A LA CULTURA AFROAMERICANA

ACTIVIDAD DEPORTIVA**:** KARATE DO PRINCIPIANTE MIXTO

ESTRATEGIAS DE LECTURA EN INGLÉS I **(**PARA OTRAS CARRERAS**)**

ÁLGEBRA LINEAL

ÁLGEBRA LINEAL

SEM. REAL. NAC.1**:** DESARROLLO COMUNAL

MÉTODOS ESTADÍSTICOS

ESTADÍSTICA COMPUTACIONAL I

MODELOS PROBABILÍSTICOS DISCRETOS

APRECIACION DE TEATRO

CÁLCULO PARA CIENCIAS ECONÓMICAS I

SEMINARIO DE REALIDAD NACIONAL I DERECHOS HUMANOS Y SEGURIDAD CIUDADANA

ESTADÍSTICA INTRODUCTORIA I

SALUD SEXUAL REPRODUCTIVA Y DERECHOS

CONTABILIDAD INTERMEDIA II

CONCEPTOS GENERALES DE AUDITORÍA I

Levels**:** ACTIVIDAD DEPORTIVA**:** KARATE DO PRINCIPIANTE MIXTO ÁLGEBRA LINEAL ... SEMINARIO DE REALIDAD NACIONAL I DERECHOS HUMANOS Y SEGURIDAD CIUDADANA

notas**$**nombre\_curso %<>% toupper**()** %>% chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

notas**$**nombre\_curso

"METODOS ESTADISTICOS"

"ESTADISTICA COMPUTACIONAL I"

"MODELOS PROBABILISTICOS DISCRETOS"

"PRINCIPIOS DE FINANZAS"

"ENTORNO LEGAL DE LAS EMPRESAS I"

"ESTRATEGIAS DE LECTURA EN INGLES I (PARA OTRAS CARRERAS)"

"ALGEBRA LINEAL"

"METODOS ESTADISTICOS"

"ESTADISTICA COMPUTACIONAL I"

"CURSO INTEGRADO DE HUMANIDADES II"

"CALCULO PARA CIENCIAS ECONOMICAS I"

"METODOS ESTADISTICOS"

"ALGEBRA LINEAL"

"SEM. DE REAL. NAC. 2: SALUD COMUNITARIA"

"LITERATURA ESPANOLA III"

"PSICOLOGIA GENERAL I"

"SEMINARIO DE REALIDAD NACIONAL I DERECHOS HUMANOS Y SEGURIDAD CIUDADANA"

"CALCULO PARA CIENCIAS ECONOMICAS I"

"SEM. REAL. NAC.1: DESARROLLO COMUNAL"

"INTRODUCCION A LA ECONOMIA"

"ESTADISTICA INTRODUCTORIA I"

"ITALIANO BASICO I"

"INTRODUCCION A LA CULTURA AFROAMERICANA"

"ACTIVIDAD DEPORTIVA: KARATE DO PRINCIPIANTE MIXTO"

"ESTRATEGIAS DE LECTURA EN INGLES I (PARA OTRAS CARRERAS)"

"ALGEBRA LINEAL"

"ALGEBRA LINEAL"

"SEM. REAL. NAC.1: DESARROLLO COMUNAL"

"METODOS ESTADISTICOS"

"ESTADISTICA COMPUTACIONAL I"

"MODELOS PROBABILISTICOS DISCRETOS"

"APRECIACION DE TEATRO"

"CALCULO PARA CIENCIAS ECONOMICAS I"

"SEMINARIO DE REALIDAD NACIONAL I DERECHOS HUMANOS Y SEGURIDAD CIUDADANA"

"ESTADISTICA INTRODUCTORIA I"

"SALUD SEXUAL REPRODUCTIVA Y DERECHOS"

"CONTABILIDAD INTERMEDIA II"

"CONCEPTOS GENERALES DE AUDITORIA I"

Como se puede notar, la variable **nombre\_curso** ya no tiene tildes o caracteres especiales, por lo que no tendría problemas en la base de datos. La siguiente variable por depurar es **nota\_ordinaria\_num**. Es importante notar que en el archivo con datos que se está usando la variable que se tiene es **nota**, por lo que se debe usar esta para poder construir **nota\_ordinaria\_num**. En este caso, se deben transformar las indicaciones especiales (RM, AP, RP…) en 0, para evitar problemas al momento de la exportación a la base de datos. En este caso, la variable de notas se toma como un factor, por lo que se debe agregar el comando **as.character** antes de **as.numeric** para evitar problemas adicionales al transformar una variable de tipo factor a una numérica. En este caso, al transformar las indicaciones especiales en números estos se convierten en NA’s, por lo que se puede indicar que se tomen los NA’s como 0, que es lo que se quiere.

notas**$**nota

1 9 1 6 8 3 6 1 0 9 4 8 2 3 0 2 8 8 RM 9 0 9 9 AP 7 7 7 2 5 RM 4 7 9 6 0 6 8 0

Levels**:** 0 1 2 3 4 5 6 7 8 9 AP RM

notas**$**nota\_ordinaria\_num **<-** as.numeric**(**as.character**(**notas**$**nota**))**

notas**$**nota\_ordinaria\_num

1 9 1 6 8 3 6 1 0 9 4 8 2 3 0 2 8 8 **NA** 9 0 9 9 **NA** 7 7 7 2 5 **NA** 4 7 9 6 0 6 8 0

notas**$**nota\_ordinaria\_num**[**is.na**(**notas**$**nota\_ordinaria\_num**)]** **<-** 0

notas**$**nota\_ordinaria\_num

2 10 2 7 9 4 7 2 1 10 5 9 3 4 1 3 9 9 0 10 1 10 10 0 8 8 8 3 6 0 5 8 10 7 1 7 9 1

Como se puede observar, ahora las notas están en una variable que se llama **nota\_ordinaria\_num** y solo contiene números del 0 al 10, que es lo que se quiere. Para terminar con las notas, es importante resaltar que este archivo de datos no contiene las variables **periodo** y **ano**, correspondientes al período y año al cual corresponden estas notas. Para efectos prácticos, se asume que estos datos corresponden al primer período del año 2017, por lo que estas variables se pueden construir de la siguiente forma.

notas**$**periodo **<-** "1"

notas**$**ano **<-** 2017

Habiendo depurado las variables del archivo, se tiene un objeto que ya puede ser exportado a la base de datos.

#### Graduados

Al igual que en los ejemplos anteriores, el primer paso es cargar los datos a ser usados para la depuración.

graduados **<-** read.csv2**(**text **=** getURL**(**"https://raw.githubusercontent.com/andresarguedas/Practica-Estudiantes/master/data/graduados%20p.csv"**))**

Revisamos los nombres de las variables del archivo para revisar que no hayan problemas.

names**(**graduados**)**

"X.U.FEFF.carne" "apellido1" "apellido2" "nombre" "ano\_juramentacion" "cod\_titulo"

"descripcion" "grado"

names**(**graduados**)[**1**]** **<-** "carne"

Para empezar, se debe verificar que la variable **carne** no tenga problemas, ya que es la variable más importante de la base de datos.

graduados**$**carne

A5555 A6666 A7777 A8888

Levels**:** A5555 A6666 A7777 A8888

Como se puede notar, no hay problemas con la variable **carne**, por lo que se puede proseguir con la depuración de las demás variables. En este caso las variables **apellido1**, **apellido2** y **nombre** se les deben realizar los mismos procedimientos, ponerlo todo en mayúscula y eliminar tildes y caracteres especiales.

graduados**$**apellido1

Corrales Jiróm Ramírez Fernández

Levels**:** Corrales Fernández Jiróm Ramírez

graduados**$**apellido1 %<>% toupper**()** %>%

chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

graduados**$**apellido1

"CORRALES" "JIROM" "RAMIREZ" "FERNANDEZ"

graduados**$**apellido2

Yglesias Mata Álvarez González

Levels**:** Álvarez González Mata Yglesias

graduados**$**apellido2 %<>% toupper**()** %>%

chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

graduados**$**apellido2

"YGLESIAS" "MATA" "ALVAREZ" "GONZALEZ"

graduados**$**nombre

Julian Mariana Daniel Juan

Levels**:** Daniel Juan Julian Mariana

graduados**$**nombre %<>% toupper**()** %>%

chartr**(**'ÁÉÍÓÚÑ','AEIOUN', .**)**

graduados**$**nombre

"JULIAN" "MARIANA" "DANIEL" "JUAN"

La siguiente variable por depurar es **fecha\_juramentacion** la cual no está presente en este archivo, sino que se tiene la variable **ano\_juramentacion**, por lo que esta primera se debe construir con base en esta. Es importante notar que **ano\_juramentacion** solamente trae el año de graduación, por lo que, por conveniencia, se pone la fecha de graduación al primero de enero de ese año. Este se hace mediante el comando **paste(“1-1-“, sep = “”)** y, posteriormente, este objeto de tipo carácter se transforma a uno de tipo fecha mediante los comandos **dmy** y **ymd** los cuales indican que el objeto está en formato DD-MM-AAAA y lo transforman en formato AAAA-MM-DD.

graduados**$**ano\_juramentacion

2015 2015 2015 2015

graduados**$**ano\_juramentacion %<>% paste**(**"1-1-", ., sep **=** ""**)**

graduados**$**ano\_juramentacion

"1-1-2015" "1-1-2015" "1-1-2015" "1-1-2015"

graduados**$**fecha\_juramentacion **<-** ymd**(**dmy**(**graduados**$**ano\_juramentacion**))**

graduados**$**fecha\_juramentacion

"2015-01-01" "2015-01-01" "2015-01-01" "2015-01-01"

Por último, se debe revisar que la variable **cod\_titulo** no tenga problemas. El comando inicial de **options(scipen = 999)** se hizo específicamente para esta variable ya que, si está escrita en forma científica al transformarlo a tipo carácter se pierde el número.

graduados**$**cod\_titulo

3304020000000 3304020000000 3304020000000 3304020000000

graduados**$**cod\_titulo %<>% as.character**()**

Teniendo todas las variables de este archivo y de los demás archivos ya depurados, se puede proceder a la exportación de estos archivos a la base de datos.

## Exportación a la base de datos

Dado que ya se hizo la depuración de los datos en los archivos de datos individuales, se procede a hacer la exportación de estos a la base de datos en PostgreSQL. El procedimiento para hacer esta exportación se presenta en R, aunque también es posible hacerlo mediante otros métodos. El primer paso es conectarse a la base de datos desde R, por lo que necesitamos cargar la librería **RPostgreSQL** y crear el objeto con la conexión a la base de datos (**conn**).

library**(**RPostgreSQL**)**

conn **<-** dbConnect**(**PostgreSQL**()**, host **=** "localhost", user **=** "postgres",

dbname **=** "estudiantes", password **=** ""**)**

En el caso anterior, en las comillas al lado de **password** se debe escribir la clave para acceder a la base de datos. Adicionalmente, hay que hacer algunos ajustes en la base de datos dado que hay información en español y las fechas son distintas, lo cual puede producir algunos problemas.

postgresqlpqExec**(**conn, "SET client\_encoding = 'windows-1252'"**)**

Es importante notar que los nombres de las tablas en la base de datos son **cursos**, **estudiantes** y **graduados**. Teniendo la conexión a la base de datos, necesitamos comparar la información que ya está en esta con la que se quiere exportar, para evitar registros duplicados. Por lo tanto, se necesita comparar la información de la base de datos con la que se quiere agregar. En R, existe la función **antijoin**, la cual hace esta comparación, por lo que se necesitan dos objetos, la información que se quiere agregar a la base de datos y la conexión a la base de datos (**conn**). El proceso de exportación a la base de datos se presenta para cada una de las tres tablas, siendo estos una extensión de los ejemplos de la sección anterior, partiendo de los archivos depurados.

### Estudiantes

Para empezar, es importante recodar que, del ejemplo de la sección 2.4.1 se tiene el objeto *est*, que contiene la información depurada de los estudiantes. El primero paso es cargar en R la información de la tabla correspondiente de la base de datos, en este caso la tabla de estudiantes.

estudiantes **<-** dbGetQuery**(**conn, "SELECT \* FROM estudiantes;"**)**

Teniendo la información proveniente de la base de datos, se procede a hacer la comparación con la información que se quiere agregar mediante la función **antijoin**.

est. **<-** anti\_join**(**est, estudiantes, by **=** "carne"**)**

En este caso, **est.** contendrá los registros del objeto **est** que no están en la base de datos (**estudiantes**), es decir, todos los registros nuevos. El siguiente paso es arreglar el objeto **est.** para que las variables estén en el mismo orden que en la base de datos, para poder así exportarlos. Este arreglo se puede hacer mediante la función **select** del paquete **dplyr**, en el cual se especifica el objeto y después los nombres de las columnas a seleccionar, en orden.

est. **<-** select**(**est., "ano\_ingreso\_ucr", "carne", "apellido1", "apellido2", "nombre", "provincia", "canton", "distrito", "genero", "fecha\_nacimiento", "nombre\_colegio", "promedio\_admision"**)**

Teniendo ya los registros con las variables ordenadas, se puede proceder a agregar los datos a la base de datos. La exportación de los datos a la base de datos se puede hacer mediante la función **dbWriteTable** con el parámetro **append = TRUE. IMPORTANTE:** el siguiente comando no se debe ejecutar con los datos del ejemplo, dado que son datos ficticios, por lo que no se deben agregar a la base de datos.

dbWriteTable**(**conn, "estudiantes", value **=** est., append **=** **TRUE**,

row.names **=** **FALSE)**

Con esto, ya se agregó la información de los estudiantes a la base de datos.

### Notas

Al igual que el ejemplo anterior, se parte de la sección 2.4.2, donde se tiene el objeto **notas** que contiene el archivo de notas depurado. Empezamos cargando los archivos de la tabla de notas a R.

n **<-** dbGetQuery**(**conn, "SELECT \* FROM cursos"**)**

Teniendo los datos de la base de datos cargados en R en el objeto **n**, procedemos a comparar los datos en **notas** con los de **n**, para así solamente agregar datos nuevos a la base de datos.

notas. **<-** anti\_join**(**notas, n, by **=** c**(**"carne", "sigla", "ano", "periodo"**))**

Por lo tanto, ahora en el objeto **notas.** se tiene los registros de **notas** que no están en la base de datos. Como se mencionó en la sección 2.2 es necesario crear la variable **id** en los datos de notas, de tal forma que **id** sea una variable secuencial. Para poder crear la variable **id** se necesita primero ver cual es el último identificador que se usó en la base de datos. Teniendo el último identificador usado, se procede a crear la variable **id** al empezar desde el identificador más grande más uno y sumándole uno a este identificador hasta que se llegue al último registro del archivo **notas**.

M **<-** dbGetQuery**(**conn, "SELECT max(id) FROM cursos"**)**

M %<>% as.numeric**()**

notas.**$**id **<-** seq**(**M **+** 1, M **+** nrow**(**notas.**))**

La variable **M** corresponde al **id** máximo de la base de datos, es decir, dado que esta variable es secuencial, el **id** de los registros nuevos debe partir desde el máximo presente en la base de datos hasta que haya uno para cada registro. Por lo anterior es que se define **notas.$id** como la secuencia desde el identificador máximo más uno hasta el máximo más la cantidad de filas, para que así sea secuencial. Ahora, dado que en la base de datos el identificador es la primera variable, entonces se deben acomodar las columnas para que la variable **id** quede de primera. Mediante la función **select** y el operador **%>%** de la librería **dplyr** se puede hacer este acomodo de forma sencilla:

notas. **<-** select**(**notas., "id", "carne", "sigla", "nombre\_curso",

"nota\_ordinaria\_num", "creditos", "periodo", "ano"**)**

Al hacer esto, el objeto **notas.** tiene lo necesario para poder agregar los registros a la base de datos, por lo que, por último, se procede a añadir estos registros. Lo anterior se puede hacer mediante la función **dbWriteTable**, usando el parámetro **append = TRUE** para que se agreguen los valores y no se cree una nueva tabla en la base de datos. **IMPORTANTE:** el siguiente comando no se debe ejecutar con los datos del ejemplo, dado que son datos ficticios, por lo que no se deben agregar a la base de datos.

dbWriteTable**(**conn, "cursos", value **=** notas., append **=** **TRUE**,

row.names **=** **FALSE)**

Con el comando anterior, ya los registros han sido agregados a la tabla **cursos** en la base de datos, por lo que ya estos pueden ser usados para el análisis de datos posterior.

### Graduados

Empezamos importando los datos de los graduados de la base de datos a R.

grad **<-** dbGetQuery**(**conn, "SELECT carne, EXTRACT(year FROM fecha\_juramentacion) AS ano FROM graduados;"**)**

Con esto, en el objeto **grad** se tienen los registros de la base de datos de graduados. El siguiente paso es comparar los registros de la base de datos con los depurados anteriormente, para agregar así solamente los registros nuevos a la base de datos.

grad. **<-** anti\_join**(**graduados, grad**)**

Por lo tanto, ahora en el objeto **grad.** se tienen los registros de **graduados** que no están en la base de datos. Ahora, al igual que en los ejemplos anteriores, ordenamos el objeto **graduados** de tal forma que las variables estén como están en la base de datos. Posteriormente, se procede a exportar estos datos a la base de datos. **IMPORTANTE:** el comando para exportar los datos a la base de datos no se debe ejecutar con los datos del ejemplo, dado que son datos ficticios, por lo que no se deben agregar a la base de datos.

grad. **<-** select**(**grad., "carne", "apellido1", "apellido2", "nombre", "fecha\_juramentacion", "cod\_titulo", "descripcion\_titulo"**)**

dbWriteTable**(**conn, "graduados", value **=** grad., append **=** **TRUE**,

row.names **=** **FALSE)**

Teniendo todos los datos exportados a la base de datos, se puede proceder al análisis de los datos.

## Análisis de los datos

El resultado final del sistema es una tabla, en la cual se presenta, para cada año (en las filas), la cantidad de personas que matricularon la carrera de Estadística (definido como la cantidad de personas que matricularon por primera vez los cursos XS-1110 o XS-0111 en ese año) y, para años posteriores (en las columnas), cuantos matricularon cursos de la carrera, hasta tener, en la última columna, la cantidad de graduados que ingresaron en ese año. Adicionalmente, en la última fila se calcula la cantidad de retiros de un año a otro, es decir, por ejemplo, la cantidad de retiros en el año 2001 es la cantidad de personas que matricularon un curso del plan de estudios de Estadística en el 2000 pero no el en 2001. A continuación, se presenta un ejemplo de la estructura de la tabla:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | Graduados |
| 2000 | - | - | - | - | - | - | - |
| 2001 |  | - | - | - | - | - | - |
| 2002 |  |  | - | - | - | - | - |
| 2003 |  |  |  | - | - | - | - |
| 2004 |  |  |  |  | - | - | - |
| 2005 |  |  |  |  |  | - | - |
| Retiro | - | - | - | - | - | - | - |

El proceso de análisis de datos, al igual que los procedimientos anteriores, se muestra en este manual mediante el uso de R, pero puede ser realizado usando otros programas o lenguajes. En este caso, en el archivo **funciones.R**, hay una función llamada **analisis**, la cual toma como argumento el nombre de la conexión a la base de datos desde R (**connection**) y devuelve la tabla con los resultados del análisis. Adicionalmente, la función crea un archivo en Excel con los resultados de la tabla y tiene dos argumentos adicionales, **inicio** y **fin,** los cuales especifican los años sobre los cuales se va a hacer la tabla. Por lo tanto, si se especifican en esos parámetros el 2010 y el 2016, la tabla resultante solamente tendría resultados correspondientes a esos dos años. Lo primero que hace la función es identificar los carnés de las personas que han matriculado alguno de los cursos introductorios de la carrera (XS-1110 o XS-0111) y el año en que lo hicieron. Luego, para cada carné identificado, se identifican los años en los que la persona matriculó los cursos y se escoge el menor de estos años, el cual se define como el año en el que la persona entró en la carrera. Posteriormente, se define la matriz que se usará para presentar los resultados finales del análisis y se agregan los nombres apropiados a las columnas y las filas. Con la matriz definida, el siguiente paso es hacer una lista, para cada persona matriculada en Estadística, con los cursos del plan de estudios que han matriculado y el año en el que lo hicieron, para así tener una lista, para cada persona, con los años en los que matriculó algún curso del plan de estudios. Con esta lista, se pueden agrupar todas las personas que matricularon Estadística en un año específico, teniendo así la cantidad de personas que matricularon cursos del plan de estudios para cada uno de los años posteriores, que es lo que se quería para la tabla final. Para los graduados, el año de ingreso se toma como el año de ingreso a la universidad (que se puede obtener del carné). Por lo tanto, con base en la información de graduados de la base de datos, se extraen todos los carnés de los graduados y, con base en el carné, se agrupan los graduados según el año de ingreso y se puede llenar la tabla con esta información. Por último, se calculan los retiros de un año a otro, los cuales se fijan en 0 para el primer año de la tabla y, para años posteriores, se compara el total de estudiantes matriculados y se calcula la diferencia entre el total de ese año y el del año anterior, sin tomar en cuenta los nuevos ingresos, es decir, se calcula la diferencia entre estudiantes que no son de primer ingreso de ese año con los que matricularon el año pasado. El resultado de la función, por lo tanto, es una matriz que contiene la información requerida para llenar la tabla de las cohortes.

Supongamos que tenemos el objeto **conn**,como se definió durante la etapa de exportación de los datos a la base de datos. Ahora, necesitamos cargar el script **funciones.R**, el cual contiene la función **analisis**, la cual hace el análisis de los datos. Este script está disponible en el siguiente link: <https://github.com/andresarguedas/Practica-Estudiantes/blob/master/funciones.R>. Por lo tanto, se necesita cargar el script desde esta dirección y cargarlo al ambiente global de R. Los siguientes comandos, adaptados de los propuestos por Petchey (2015, parr. 9), permiten cargar el script desde GitHub, usando la librería **RCurl**:

library**(**RCurl**)**

script **<-** getURL**(**"https://raw.githubusercontent.com/andresarguedas/Practica-Estudiantes/master/funciones.R", ssl.verifypeer **=** **FALSE)**

eval**(**parse**(**text **=** script**))**

Teniendo el script cargado, se puede usar la función **analisis**, por lo que, para hacer el análisis de los datos, lo que se necesita hacer es lo siguiente:

analisis**(**connection **=** conn**)**

Un resultado de la función se presenta a continuación:

2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 Graduados

2000 48 17 14 9 7 6 6 5 4 2 1 2 2 2 0 0 0 0 17

2001 0 20 11 8 5 4 4 3 1 1 2 1 0 0 0 0 0 0 9

2002 0 0 25 13 7 7 3 2 2 1 0 2 1 0 0 1 1 1 16

2003 0 0 0 25 17 12 10 9 5 3 2 1 1 1 0 1 0 1 12

2004 0 0 0 0 17 15 13 12 9 6 5 3 2 1 0 0 0 0 12

2005 0 0 0 0 0 26 12 7 6 2 3 4 1 1 0 2 0 0 9

2006 0 0 0 0 0 0 36 26 18 13 9 6 5 0 1 3 0 0 11

2007 0 0 0 0 0 0 0 38 24 15 17 11 11 10 8 9 6 4 12

2008 0 0 0 0 0 0 0 0 36 23 22 18 13 12 10 8 3 2 10

2009 0 0 0 0 0 0 0 0 0 46 33 29 30 20 14 10 8 6 14

2010 0 0 0 0 0 0 0 0 0 0 56 48 37 28 23 22 14 8 10

2011 0 0 0 0 0 0 0 0 0 0 0 64 49 39 38 31 23 18 10

2012 0 0 0 0 0 0 0 0 0 0 0 0 76 55 52 43 29 20 10

2013 0 0 0 0 0 0 0 0 0 0 0 0 0 93 70 55 47 29 5

2014 0 0 0 0 0 0 0 0 0 0 0 0 0 0 120 100 82 64 0

2015 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 85 70 57 0

2016 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 101 83 0

2017 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 103 0

Retiro 0 31 12 20 19 9 22 20 33 39 18 25 37 59 46 51 87 91 0

Como se comentó anteriormente, esta es la tabla de cohortes que se quería, por lo que la función da como resultado lo que se esperaba. Otra opción para cargar el script es acceder directamente a la dirección dada anteriormente y copiar el contenido del archivo a un nuevo script en R y correrlo, lo que daría los mismos resultados a los que se presentaron anteriormente. Por lo tanto, lo único que faltaría es darle formato a la tabla para poder presentarla, dado que los datos requeridos para su construcción ya se obtuvieron mediante la función **analisis**.

**Figura 1**

**Ejemplo de carta enviada a la Oficina de Registro e Información para la petición de los datos**

Imagen que contiene texto

Descripción generada con confianza muy alta