TRABAJO DE FIN DE GRADO

Grado en Estadística

Título: El uso del “Propensity score” en estudios observacionales: estimación e ilustración con datos reales

Autor: Oriol Planesas Pérez

Director: Lesly María Acosta Argueta

Departamento: Universitat Politècnica de Catalunya. Departament d'Estadística i Investigació Operativa

Convocatoria: Junio 2021

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Resumen

El propósito de este TFG es el de estudiar el *propensity score* (PS) y sus diversos métodos sobre un estudio observacional para estimar el efecto de un tratamiento o similar sobre una variable respuesta. La idea del uso del PS es reducir el llamado sesgo de confusión y poder así tratar un estudio observacional como si fuese aleatorizado. Este trabajo está dividido en 5 capítulos. En el primero introduciremos el TFG. En el segundo aprenderemos qué es el *propensity score*, como se utiliza este en un estudio observacional y para qué y veremos cómo se calcula esta probabilidad condicional. También describiremos los 4 diferentes métodos basados en el uso del *propensity score*. En el tercero ilustraremos cómo aplicar estos métodos a una base de datos y cómo implementarlos mediante el software estadístico R, a la vez que comentaremos los resultados de esta base de datos y cómo interpretarlos. En el cuarto capítulo, con dos de los métodos basados en PS y una base de datos más reciente, estudiaremos el efecto de una variable tratamiento sobre una variable respuesta. Finalmente, en el quinto presentaremos las conclusiones, discusión y limitaciones de este TFG.

Palabras clave

Propensity score, modelo lineal generalizado, estudios observacionales, regresión logística, variable respuesta, variable tratamiento.

Clasificación AMS

62-07 Data analysis

62J12 Generalized linear models

62P10 Applications to biology and medical sciences

Agradecimientos

Agradecer a Lesly por toda su ayuda y consejos que me ha proporcionado mientras realizaba este trabajo de fin de grado. También agradecer a mi familia y amigos más cercanos por todo el apoyo que me han dado.

Índice

Lista de figuras

Lista de cuadros

Lista de ecuaciones

Capítulo 1

INTRODUCCIÓN

En un estudio aleatorizado, la asignación a grupos tratamiento se hace siempre al azar y por tanto se puede garantizar que el efecto sobre la variable respuesta de interés se debe solo a la variable tratamiento.

Para un estudio para estimar el efecto de un tratamiento sobre una variable respuesta sería deseable usar un diseño de estudio aleatorizado, pero ello no siempre es factible, sea por logística o razones éticas.

La alternativa es utilizar estudios observacionales para estimar el efecto de un tratamiento sobre una variable respuesta. El problema de estos es el llamado sesgo de confusión, debido a que pueden existir otras covariables (de confusión) que pueden distorsionar el efecto del tratamiento; es decir, no puede decirse como en los estudios aleatorizados que el efecto observado se deba solo a la variable tratamiento.

El propósito de este trabajo final de grado (TFG) es el de estudiar el *propensity score* (PS) y sus diversos métodos sobre un estudio observacional para estimar el efecto de un tratamiento o similar sobre una variable respuesta de interés. La idea del uso del PS es reducir el llamado sesgo de confusión y poder así tratar un estudio observacional como si fuese aleatorizado (RCT).

Rosenbaum y Rubin en 1983 [1] introdujeron el concepto del PS, explican que es y cómo utilizarlo, además de explicar los diferentes ámbitos donde se puede usar. En 2011, Austin [2] actualizó

este concepto, explorando más los diferentes métodos que se basan en su uso y los comparó con resultados de un estudio de regresión clásico.

Este trabajo está dividido en 4 capítulos, aparte de la introducción. En el segundo capítulo, explicaremos como calcular el PS y como se usa en un estudio observacional y para qué; también describiremos cuatro diferentes métodos basados en el PS. En el tercero ilustraremos cómo aplicar estos métodos a una base de datos y cómo implementarlos mediante el software estadístico R, a la vez que comentaremos los resultados de esta base de datos y cómo interpretarlos. En el cuarto capítulo, con dos de los métodos basados en PS y una base de datos más recientes, estudiaremos el efecto de una variable tratamiento sobre una variable respuesta. Finalmente, en el quinto presentaremos las conclusiones, discusión y limitaciones de este TFG.

Capítulo 2

METODOLOGÍA

En este capítulo introduciremos el concepto de *propensity score* (PS), cómo calcularlo con regresión logística y describiremos los diferentes métodos para estimar el efecto de la variable tratamiento sobre una variable respuesta. Además comentaremos algunos aspectos clave a tomar en cuenta cuando realizamos un estudio basado en PS. También explicaremos qué son las variables de confusión y cómo identificarlas.

2.1. Introducción al *propensity score*

Cuando queramos estudiar el efecto de un tratamiento necesitamos comparar los resultados obtenidos con los individuos con el tratamiento y con aquellos sin el tratamiento. A estos los llamaremos grupo tratamiento y grupo control, respectivamente. Si comparamos los resultados de dichos grupos, podremos observar los efectos que puede generar el tratamiento sobre los pacientes. Esto lo estimaremos con el promedio del efecto del tratamiento (ATE) [3]; comparamos la diferencia entre medias en los resultados entre unidades asignadas al grupo tratados ($Y_i = 1$) y no tratados ($Y_i = 0$):

$$\text{ATE} = E[(Y_i = 1) - (Y_i = 0)] = E[Y_i = 1] - E[Y_i = 0] \tag{2.1}$$

Para estudiar el efecto de un tratamiento, lo ideal sería utilizar estudios aleatorizados para evitar el sesgo de confusión. Un estudio aleatorizado consiste en un diseño de estudio, donde la asignación de tratamiento a los individuos se realiza al azar. Se espera así poder garantizar que cualquier diferencia significativa entre estos grupos respecto a una variable respuesta se deba solo a la variable tratamiento y no a otras variables. No obstante, en la práctica, esto no es siempre posible por razones logísticas ó éticas. El problema de los estudios observacionales es que no solo está la variable tratamiento, sino que existen más variables que pueden afectar a nuestro estudio. A raíz de esto, no sabremos diferenciar si el efecto del tratamiento proviene del propio tratamiento o de estas otras variables. Para controlar este sesgo generado en los estudios observacionales, podemos utilizar los métodos de PS.

El *propensity score*, fue introducido por primera vez en 1983 por Rosenbaum y Rubin [1] y su uso para el control de los factores de confusión se ha ido incrementando recientemente.

Definición de *propensity score*

Este es un resumen de todas las medidas de las características del paciente previas al tratamiento, es decir, es una probabilidad condicionada de ser asignado a un tratamiento en particular dado un vector de covariables X_i observadas:

$$e_i = e(X_i) = P(Y_i = a \mid X_i) \tag{2.2}$$

Siendo i el individuo a calcular su *propensity score* sobre la variable tratamiento Y_i. También tenemos que a, que corresponde al valor que toma la variable tratamiento para el individuo i, puede tomar los valores 0 y 1, dependiendo si el paciente es del grupo control o tratamiento, respectivamente.

Variable de confusión

Una variable de confusión es aquella que cumple los requisitos que se indica en la figura 2.1. Es una variable que afecta tanto a la variable tratamiento y variable respuesta y puede afectar a como

vemos el efecto del tratamiento sobre la variable respuesta. Podemos poner como ejemplo cómo el BMI de un individuo puede ser variable de confusión entre el efecto que puede tener la cantidad de alcohol que toma una persona sobre la mortalidad.

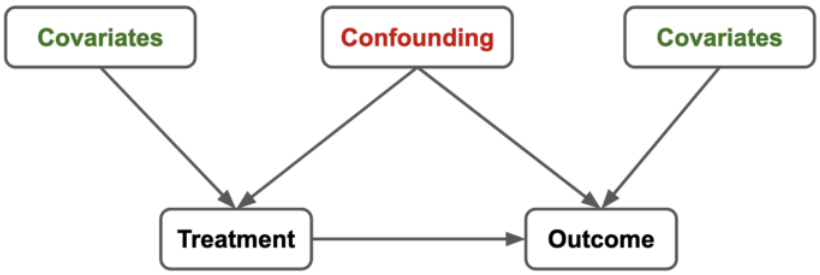


Figura 2.1: Descripción gráfica del efecto de una variable de confusión. [4]

En la Figura 2.1 podemos ver cómo sería una representación gráfica de como afecta una variable de confusión a una variable dependiente y a una variable independiente, y cómo puede distorsionar la primera a las otras.

2.2. Estimación del *propensity score*

Los valores de *propensity score* para cada individuo se calculan utilizando una regresión logística *logit* sobre la variable tratamiento.

$$\ln \frac{e(X_i)}{1-e(X_i)} = b^T \cdot X_i \tag{2.3}$$

Donde:

$$e(X_i) = P(Y_i = 1 \mid X_i) = \frac{e^{b^T \cdot X_i}}{1+e^{b^T \cdot X_i}} = \frac{1}{1+e^{-b^T \cdot X_i}} \tag{2.4}$$

Tenemos que $e(X_i)$ es el valor de PS para cada individuo i y $b^T \cdot X_i$ son los coeficientes del modelo multiplicado por los valores que toman las variables para cada individuo i. La $P(Y_i = 1 \mid x_i)$ solo puede tomar valores entre 0 y 1. En el apartado 3.2 veremos cómo se calcula en R.[1]

2.3. ¿Qué variables escogemos para el estudio del PS?

Desde hace años está el debate sobre qué variables debemos escoger para realizar el estudio de *propensity score* sobre el tratamiento. El principal motivo de existencia del PS es eliminar una posible variable de confusión de un estudio observacional, por eso, podríamos hacer que las variables a seleccionar serán aquellas que veamos que pueden ser este tipo de variable. También podemos simplemente coger todas las variables del estudio debido a que cuando hacemos métodos como el *matching*, que explicaremos en el apartado 2.4.1, todas las posibles variables de confusión que existen desaparecerían. Todo depende de cómo se quiera enfocar el estudio.

Austin, Grootendorst y Anderson [5] examinan los beneficios que puede suponer utilizar las variables de diversas maneras para estudiar el PS. Entre ellas se encuentra el hecho de que, si utilizamos cualquier combinación de las variables del estudio, hace que los individuos de ambos grupos, control y tratamiento, pasen a estar balanceados pero aumentando la varianza y sesgo del estudio. Si utilizamos solo las variables de confusión, previamente deberíamos realizar un estudio para ver cuáles de ellas lo son.

En este trabajo, para el cálculo del PS usaremos solo las variables de confusión.

[1] No confundir la $e(X_i)$ que corresponde a nuestro valor de *propensity score* con e^x que corresponde al número de Euler. Donde aquí x corresponde al valor que puede tener un paciente del estudio en el modelo logístico.

2.4. Métodos de uso del *propensity score* para estimar efectos de tratamiento

Teóricamente, el PS se puede aplicar a los estudios aleatorizados (RCT) y a los estudios observacionales. En un RCT balanceado con una variable tratamiento con 2 categorías, los valores del PS para cada individuo siempre será 0.5. En cambio, en los estudios observacionales necesitamos estimar el valor de PS, como hemos explicado en el apartado 2.2, que en este trabajo se hará con un modelo de regresión logística.

Austin (2011) [2] propuso cuatro métodos distintos de PS para mitigar los efectos de confusión en un estudio. Estos métodos son: PS *matching*, *Inverse probability of Treatment Weighting estimation* (IPW), estratificación y ajuste de regresión de covariables con el PS. A continuación, describiremos cada uno de estos métodos.

2.4.1. PS *matching*

El objetivo de este método es producir una distribución balanceada de las covariables según el grupo tratamiento. De esta manera, el estudio observacional emula un diseño RCT y reduce el sesgo producido por la presencia de variables de confusión. Los individuos se emparejan en base al PS calculado previamente.

Técnicas para PS *matching*

Primero hemos de decidir si utilizaremos o no reemplazamiento. Rosenbaum (2002) [6] explica que la diferencia entre estos dos consiste en que, si trabajamos con reemplazamiento, cada observación del grupo control puede utilizarse una o más veces a la hora de emparejar con la variable tratamiento. Si no utilizamos reemplazamiento, cada observación solo se podría utilizar una vez.

Rosenbaum [6] propone otra forma de hacer PS *matching*: técnica avariciosa. La primera consiste en escoger aleatoriamente un individuo del grupo tratamiento y luego, del grupo control escoger aquel que tenga un PS más cercano, y así hasta que todos los sujetos tengan pareja. En contraste, está el óptimo: escoge un individuo de cada grupo para minimizar la diferencia entre los valores de

PS entre dos individuos.

Rosenbaum y Rubin (1985) [7] proponían diversos métodos para hacer el PS *matching*: emparejando por vecino más cercano y emparejando por vecino más cercano dentro de una distancia específica llamada distancia de Caliper [8]. El primero consiste en seleccionar las parejas por cómo de cerca están sus valores de PS. Si hay múltiples valores cercanos entre los grupos, se selecciona uno al azar. En el otro método se añade una restricción que consiste en que la diferencia entre los PS de cada individuo no puede sobrepasar un umbral: la distancia de Caliper.

Una vez formado este emparejamiento, el efecto del tratamiento puede ser estimado comparando los resultados entre los pacientes del grupo tratamiento y del grupo control en el *matching sample*. Si la variable respuesta es una variable continua, utilizaremos las diferencias entre el promedio de cada grupo. Si esta es dicotómica, en vez de promedios compararemos las proporciones de cada grupo.

Una vez tengamos la estimación del PS, podremos estimar el efecto del tratamiento. El saber cómo estimar el PS paso por paso será explicado en apartados posteriores.

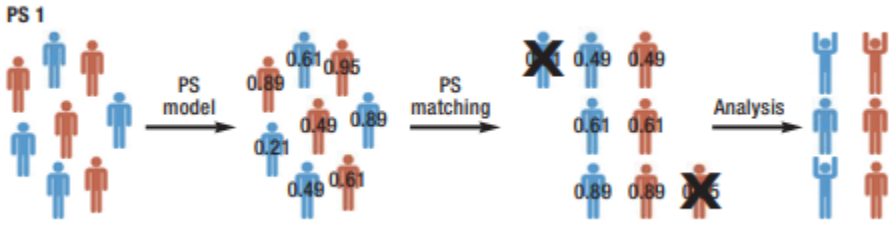


Figura 2.2: Representación gráfica del estudio del PS *matching* [9].

En la Figura 2.2 podemos ver cómo sería un ejemplo de separar individuos por grupos en este método de estudio. Realizamos el cálculo del PS por regresión logística para cada individuo. Utilizamos alguno de los métodos de emparejamiento mencionados, descartamos los individuos no emparejados y con los individuos restantes realizaremos el análisis final.

2.4.2. *Inverse Probability of treatment Weighting estimation*

Este método utiliza ponderaciones basadas en el PS para crear muestras que en cada distribución de covariables sea independiente de cómo se asigne el tratamiento.

Si tomamos Y_i como una variable indicadora de si el sujeto i es tratado o no y $e(X_i)$ el PS calculado previamente, la ponderación puede ser definida como:

$$W_i = \frac{Y_i}{e(X_i)} + \frac{1 - Y_i}{1 - e(X_i)} \tag{2.5}$$

La ponderación de un sujeto es igual a la probabilidad inversa de recibir dicho tratamiento.

Esta fue introducida por Rosenbaum (1987) [10] como una forma de modelo directamente basado en estandardización.

Estas ponderaciones pueden ser poco precisas o inestables si los sujetos tienen poca probabilidad de recibir el tratamiento. En el 2000, Robins, Hernan y Brumback [11] propusieron un método para arreglar este inconveniente utilizando ponderaciones estabilizadoras. Según Austin [2], en 2008, Morgan y Todd [12] estimaron el ATT (*Average effect in Treatment in Treat group*) y el ATC (*Average effect in Treatment in Control group*) con estas ponderaciones:

$$W_i = \frac{1}{e_i}[\text{ATT}] \tag{2.6}$$

$$W_i = \frac{1}{1 - e_i}[\text{ATC}] \tag{2.7}$$

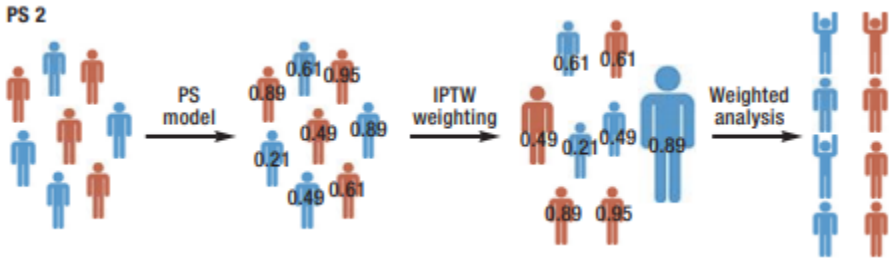


Figura 2.3: Representación gráfica del PS IPW [9].

En la Figura 2.3 vemos cómo se realizaría un estudio del PS con el método de IPW. Como en el anterior, primero calcularíamos los valores del PS para cada individuo, después realizaríamos el estudio de ponderaciones, donde algunos individuos representan más un grupo que otro dependiendo del valor PS, y añadiríamos estos valores a la base de datos original.

2.4.3. Estratificación

La estratificación en PS implica estratificar a los sujetos en subconjuntos exclusivos basados en el PS estimado. Primero se clasifica a los sujetos dependiendo de su PS estimado, para posteriormente ser estratificados en subgrupos basados en umbrales previamente definidos.

Cochran (1968) [13] demostró que si dividimos los sujetos en 5 subgrupos eliminamos aproximadamente el 90 % del sesgo debido a la variable de confusión. Más tarde, Rosenbaum y Rubin [1] extendieron este resultado al PS, eliminando también el 90 % del sesgo para el PS. Cuando el PS sea especificado correctamente, la distribución de las covariables medidas será aproximadamente similar entre sujetos tratados y no tratados dentro de un mismo estrato.

Con cada estrato, el resultado de los efectos del tratamiento puede ser estimado comparando directamente los resultados entre el grupo control y el grupo tratamiento.

Figura 2.4: Representación gráfica del PS estratificado [9].

En la Figura 2.4 vemos cómo sería el estudio del PS por estratificación. Primero calcularíamos los valores PS para cada individuo y después separaríamos por los estratos que sean convenientes a nuestros individuos. Finalmente, comprobaríamos que, para cada estrato, hemos eliminado las posibles variables de confusión.

2.4.4. Ajuste de regresión de covariables para el PS

Realizamos un modelo de regresión teniendo en cuenta las variables que consideramos de confusión y la variable tratamiento como variable dependiente con un modelo de regresión logística. De aquí sacaremos los valores PS con los valores *fitted* del modelo.

Cuando tengamos hecho este modelo, añadiremos los valores del PS a la base de datos como si fuera una covariable más. Después realizaremos otro modelo con la variable respuesta como variable dependiente y como variables independientes las variables tratamiento y los valores PS.

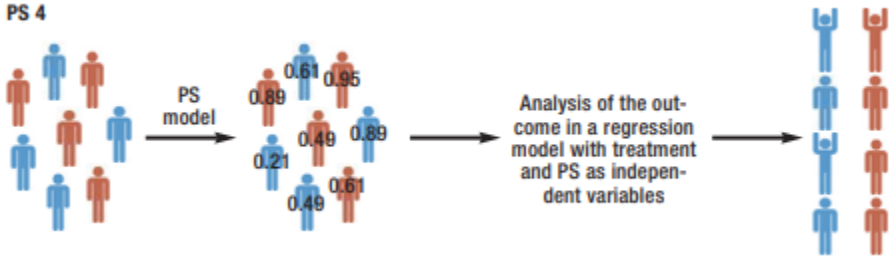


Figura 2.5: Representación gráfica del estudio del PS por regresión de covariables [9].

En la Figura 2.5 vemos como sería el estudio del PS por ajuste de regresión de covariables. Como en todos los demás, realizaremos el cálculo del PS para cada individuo y veremos si estos valores son significativos.

2.5. Comparación de los diferentes métodos de PS

Austin (2011) [2] compara estos diferentes métodos debido a que es más común utilizar el ajuste de regresión que los métodos de PS. Por eso, sería recomendable mirar las ventajas y desventajas de cada forma de estudiar el efecto del tratamiento.

El método más común es el PS *matching* gracias a su sencillez debido a que lo que debemos realizar es emparejar individuos y a su efectividad comparándolo con los demás métodos. No obstante, tiene el inconveniente que esta elimina los individuos que no pueden ser emparejados si utilizamos el método más habitual dentro del PS *matching*: el método simple, el cual empareja un individuo con otro del otro grupo y elimina de la nueva base de datos los individuos que no pueden ser emparejados. Sin embargo, asegura que de esta salen unos datos correctos para ser estudiados con el PS. Este método también restringe el hecho que ambos grupos han de tener el mismo número de pacientes, cosa que con el IPW o con el estratificado no sucede.

Si buscamos utilizar el método más sencillo de todos y el más rápido de realizar podemos utilizar el ajuste de regresión de covariables. Lo único que necesitamos calcular es los valores del PS del modelo realizado e introducirlo como una variable independiente más junto a la variable tratamiento. De hecho, el PS *matching* y la regresión de covariables en base al PS son los dos métodos más utilizados en la investigación biomédica.

Oliver Kuss, Maria Blettner y Jochen Börgermann (2016) [9] resumen la comparación entre los diversos métodos dependiendo de si se quiere sacrificar el sesgo que nos saldrá en el estudio o la precisión del estimador. Si utilizamos el PS *matching* tendremos un sesgo más bajo, pero al perder individuos, la estimación del estudio será menos eficiente.

2.6. Herramientas de verificación de balance de distribución de covariables

Después de realizar el análisis de los métodos basados en el uso del PS, necesitaremos comprobar que el balance de distribución de covariables es correcto. Para ello observaremos la diferencia de promedios estandardizada (SMD) por los grupos generados con la variable tratamiento. Si vemos que este valor es lo suficientemente pequeño, entonces no existen diferencias entre los diferentes grupos tratamiento. En nuestro estudio sería comparar los valores obtenidos para el grupo control y tratamiento, y ver si existen diferencias entre estos grupos después de realizar los diferentes métodos. Nosotros tendremos en cuenta la regla de dedo (rule of thumb) que usa como valor umbral el 0.1, donde un SMD menos a 0.1 indicaría una distribución de covariables no balanceada según el grupo tratamiento (hay diferencia entre grupos). Este proceso es idéntico al de identificar una variable de confusión pero con los nuevos datos conseguidos con los métodos basados en el uso del PS.

También, podemos utilizar diversos tipos de gráficos, como pueden ser los gráficos de barras apiladas, que nos muestra como de balanceadas están las covariables según los grupos tratamiento o estratos.

2.7. Pasos a seguir para el estudio

Para realizar el estudio del PS necesitaremos realizar los pasos mencionados en un orden concreto, que pondremos en práctica cuando lo apliquemos en R. Todo lo que explicaremos a continuación se representa en la Figura 2.6. Estos pasos son:

- Realizar un análisis descriptivo de la base de datos y estudiar cómo son las variables que poseemos en los datos para así saber qué tenemos y cómo vamos a realizar el estudio observacional. Con esto también debemos comprobar qué variables son de confusión y cuáles no lo son.

- A continuación, calcular el PS para cada individuo.

- Se escoge uno de los 4 métodos basado en el uso del PS.

- Finalmente, realizar un estudio del balance de los datos para comprobar que se han eliminado las variables de confusión. Esto se aplica a todos los métodos excepto el de regresión por covariables.

Con todo esto, ya podemos realizar el análisis de la base de datos resultante con el uso del PS.

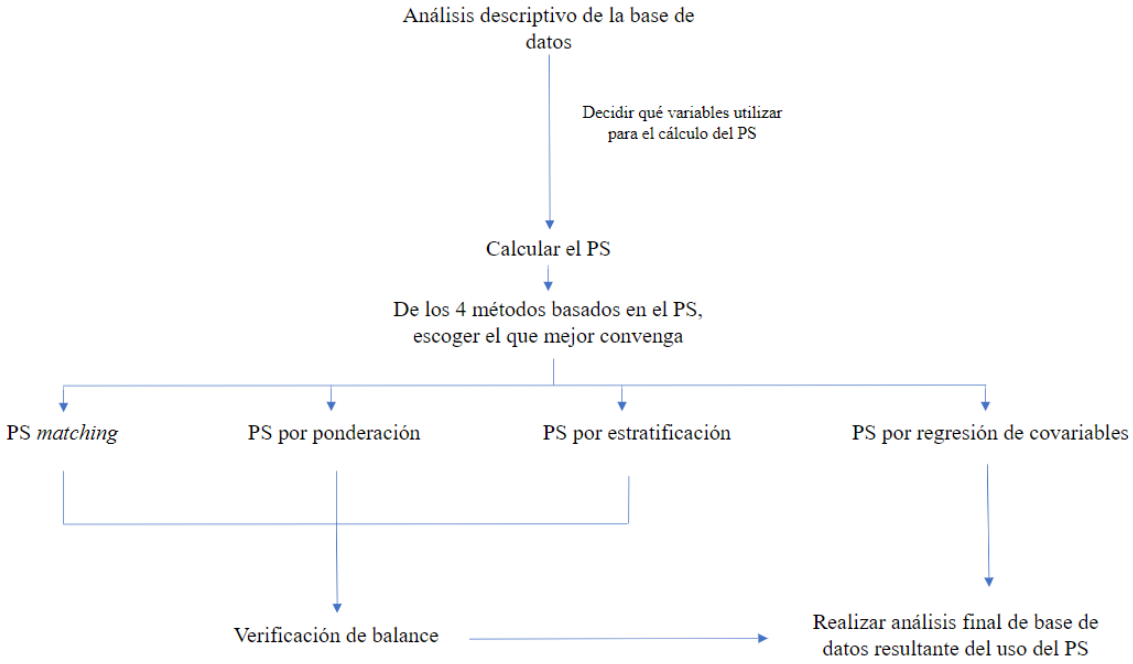


Figura 2.6: Esquema de cómo se realizaría un análisis de datos basado en el uso del PS.

Capítulo 3

ILUSTRACIÓN DEL USO DEL PS EN R

En este capítulo, ilustraremos como implementar en R cada uno de los 4 métodos basados en el uso del PS descritos en el capítulo anterior. Esto conlleva a explicar las funciones de R, los paquetes que utiliza y cómo analizar los resultados obtenidos. Para ello, usaremos la base de datos incluida en el paquete *MatchIt* [14].

3.1. Descripción de la base de datos Lalonde

Esta base de datos, presente en el paquete de R *MatchIt*, consiste en una muestra extraída de una base de datos de un estudio de la *National Supported Work Demonstration* (NSWD) del grupo tratamiento y una muestra del grupo control del *Population Survey of Income Dynamics* (PSID).

La base de datos Lalonde contiene datos de 614 individuos, de los cuales 185 (30.1 %) han sido tratados y 429 (59.9 %) no; consta de 5 variables continuas y 2 categóricas dicotómicas. En el listado que viene a continuación describimos cada una de ellas y su descriptiva univariante en el Cuadro 3.1:

- *treat*: es la **variable tratamiento** en nuestro estudio. Consiste en la asignación de si un paciente es tratado o no. Es una variable dicotómica con valores 1 = Tratado y 0 = No tratado. Como

se ha mencionado anteriormente, 185 individuos tienen valor 1 y 429 tienen valor 0.

- *age*: edad del individuo en años. El individuo con menor edad tiene 16 años y el mayor tiene 55 años. El promedio de la edad de los individuos es de 27.36 años,donde la mayoría son jóvenes y menores de 55 años.

- *educ*: número de años que el individuo ha sido estudiante. El individuo con menos años siendo educado es de 0 años y el que más se ha educado corresponde a 18 años. El promedio de años educándose es de 10.27 años.

- *married*: indica si el individuo está casado o no; variable dicotómica con valores 1 = Casado y 0 = No casado. Hay 255 individuos que están casados y 359 que no lo están.

- *nodegree*: indica si el individuo tiene el graduado en la secundaria con una variable dicotómica con valores de 1 = No graduado y 0 = Graduado. Hay 387 individuos que no tienen la secundaria y 227 que sí la tienen.

- *re74*: ingresos de cada individuo en 1974, en dólares estadounidenses. En esta variable continua hay individuos con ingresos nulos, es decir, con valor 0, y el salario máximo se sitúa en 35040 dolares. El promedio de estos es de 4558 dólares anuales.

- *re75*: ingresos cada individuo en 1975, en dólares estadounidenses. En esta variable continua hay individuos con ingresos nulos, es decir, con valor 0, y el salario máximo se sitúa en 25142.2 dolares. El promedio de estos es de 2184.9 dólares anuales.

- *re78*: ingresos cada individuo en 1978, en dólares estadounidenses. En esta variable continua hay individuos con ingresos nulos, es decir, con valor 0, y el salario máximo se sitúa en 60307.9 dolares. El promedio de estos es de 6792.8 dólares anuales. Esta variable es la **variable respuesta** ya que al final estudiaremos el efecto sobre ella de la variable tratamiento.

treat	age	educ	married	nodegree	re74	re75	re78
0:429	Min. :16.00	Min. : 0.00	0:359	0:227	Min. : 0	Min. : 0.0	Min. : 0.0
1:185	1st Qu.:20.00	1st Qu.: 9.00	1:255	1:387	1st Qu.: 0	1st Qu.: 0.0	1st Qu.: 238.3
	Median :25.00	Median :11.00			Median : 1042	Median : 601.5	Median : 4759.0
	Mean :27.36	Mean :10.27			Mean : 4558	Mean : 2184.9	Mean : 6792.8
	3rd Qu.:32.00	3rd Qu.:12.00			3rd Qu.: 7888	3rd Qu.: 3249.0	3rd Qu.:10893.6
	Max. :55.00	Max. :18.00			Max. :35040	Max. :25142.2	Max. :60307.9

Cuadro 3.1: Descripción univariante de los datos Lalonde.

3.2. Detección de variables de confusión

Antes de realizar todo el proceso de estudio basado en el uso del PS, analizaremos que variables son de confusión. Para ello debemos ver que variables afectan tanto a la variable respuesta *re78* como a la variable tratamiento *treat*. Las diferencias entre estratos en la variable tratamiento las veremos realizando la función *CreateTableOne* del paquete *tableone* [15], que nos puede generar una tabla que nos ayuda a comparar dos grupos estratificados, es decir, un estudio bivariante de la variable *treat*. Para cada variable nos muestra el promedio, la desviación estándar de cada estrato *treat* y las diferencias de los promedios estandardizadas:

```
                     Stratified by treat
                      0                1                SMD
n                        429              185
age (mean (SD))        28.03 (10.79)    25.82 (7.16)     0.242
educ (mean (SD))       10.24 (2.86)     10.35 (2.01)     0.045
married (mean (SD))     0.51 (0.50)      0.19 (0.39)     0.719
nodegree (mean (SD))    0.60 (0.49)      0.71 (0.46)     0.235
re74 (mean (SD))     5619.24 (6788.75) 2095.57 (4886.62) 0.596
re75 (mean (SD))     2466.48 (3292.00) 1532.06 (3219.25) 0.287
```

Figura 3.1: Descriptiva bivariante de los datos Lalonde estratificada por la variable *treat*.

Para saber si existen diferencias entre los dos grupos hemos de ver como el valor de SMD[1] supera un cierto valor para nuestro estudio. Tomaremos un valor de SMD superior a 0.1 como valor umbral. Para la variable *age* vemos como existe una diferencia entre los diferentes grupos debido

[1] Página 49 en los apéndices

a que el SMD es superior a 0.1, siendo en este caso 0.242. La única variable que no tiene una diferencia es la variable *educ*, con un valor de 0.045. Donde vemos la mayor diferencia es en la variable *married*, donde su SMD es de 0.719. Tanto en la variable *re74* y *re75* vemos como en el grupo tratamiento el salario anual promedio es mucho menor que el del grupo control.

Para estudiar que covariables están relacionadas con la variable *re78* se ha realizado un modelo lineal para cada variable independiente por separado. Al realizar esto, observamos como todas las variables afectan significativamente a la variable *re78*. Como todas las variables independientes excepto *educ* afectan tanto a la variable respuesta como a la variable tratamiento, las covariables *age*, *married*, *nodegree*, *re74* y *re75* son posibles variables de confusión en nuestro estudio. Así pues, estas variables serán las que utilizaremos en el estudio basado en el uso del PS.

3.3. Cálculo del PS

En este apartado veremos cómo calcular, vía regresión logística, el valor del Ps para cada individuo en R. Para ello, se pueden utilizar diferentes funciones de diversos paquetes, como por ejemplo: *pscore* del paquete *nonrandom* o con los valores PS generados en la función *matchit* del paquete *MatchIt*, que en la función los nombra como *distance*. Este último paquete lo estudiaremos con más profundidad en el apartado 3.4. Nosotros utilizaremos los valores PS que se generan en el modelo de regresión logística que corresponden a los valores *fitted*. Podemos representar estos valores con una gráfica de densidad para ver cómo se distribuyen estos.

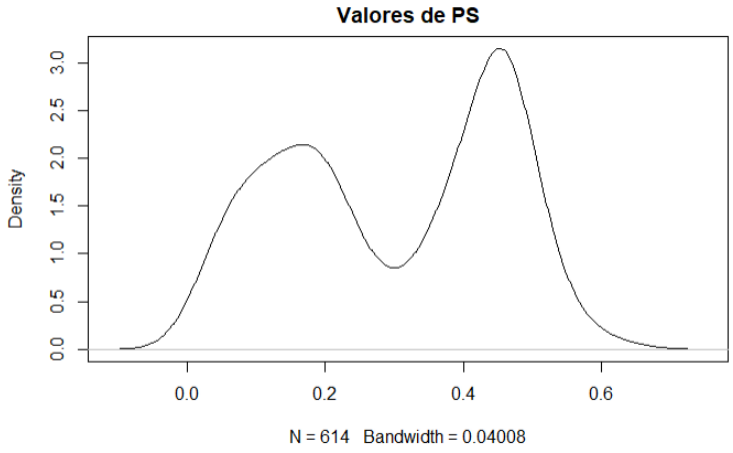
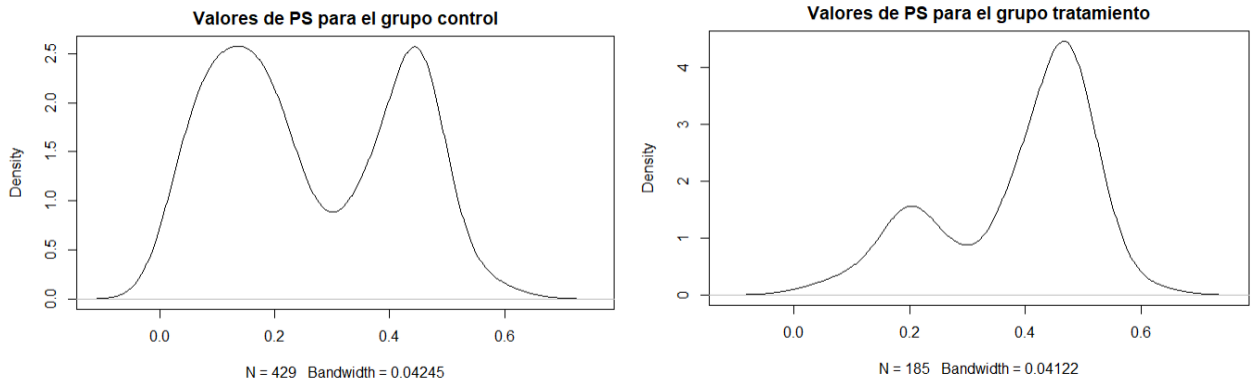


Figura 3.2: Valores del PS de los individuos de la base de datos Lalonde.

En la Figura 3.2[2] podemos observar como se distribuyen los valores del PS. Podemos ver como hay 2 picos sobre los valores 0.15 y 0.45. Esto puede deberse a que cada pico corresponde a los valores que recoge cada grupo. Para ver mejor esto separaremos los valores del PS para cada grupo.



(a) Grupo Control **(b)** Grupo tratamiento

Figura 3.3: Gráficas de densidad del PS.

[2]Página 51 de apéndices

En la Figura 3.3[3] observamos como el primer pico de valores PS se centran casi todos en el grupo control y el segundo pico lo podemos encontrar en individuos de los dos grupos.

	Promedio	Desviación estándar
PS	0.30	0.16
PS Control	0.26	0.16
PS Tratamiento	0.39	0.13

Cuadro 3.2: Valores promedio y desviación estándar para los valores del PS y estratificados por la variable tratamiento.

En el Cuadro 3.2 vemos el promedio y desviación estándar de todos los valores PS y los valores estratificados por la variable tratamiento (PS Control y PS Tratamiento). Los valores promedio PS para todos los individuos es de 0.3 con una desviación estándar de 0.16. Si miramos la Figura 3.2 vemos como el promedio se encuentra justo en el valle de los datos. Algo muy parecido pasa con los valores de PS para el grupo control, que el promedio, 0.26, se encuentra justo entre los dos picos. En cambio, los valores de PS del grupo tratamiento tienen un promedio más elevado, 0.39, y una desviación más pequeña, 0.13. Esto se debe a que, como hemos visto en la Figura 3.3, los valores se concentran más en un punto en concreto.

Ya tenemos calculados estos valores, pero para adaptar bien cada método y como los estamos ilustrando por separado, los volveremos a calcular para cada método adaptándolo de la mejor forma posible a cada uno de ellos.

3.4. PS *matching*

En este apartado ilustraremos cómo es el método del PS *matching* en R. Utilizaremos principalmente el paquete *MatchIt* [14]. Al tener ya realizado un análisis descriptivo de la base de datos, lo primero que haremos para este tipo de estudio es generar una fórmula para el modelo que se

[3]Página 51 de apéndices

estudiará, para así posteriormente añadirla en la función *matchit* del paquete mencionado anteriormente. En este caso, nuestra fórmula seria: "$treat \sim age + married + nodegree + re74 + re75$".

Esta realizará el cálculo automático del PS con regresión logística para cada individuo teniendo en cuenta la fórmula que implementaremos en esta, generando un análisis descriptivo de este, tanto para la base de datos completa cómo para solo los individuos que quedarán emparejados. Para implementarlo, debemos aplicar a la función *matchit* la fórmula realizada previamente, la base de datos que queremos estudiar y el método del PS *matching* a realizar. También podemos añadir el ratio de emparejamiento que deseamos utilizar, es decir, el número de individuos máximos que puedes emparejar con otro individuo. Una de las opciones que le añadiremos a la función será la de *nearest* en el tipo de método del PS *matching* a realizar. Esta hace que el estudio que se realiza en el emparejamiento sea la de tipo por vecino más cercano que, como hemos mencionado en el apartado 2.4 del PS *matching*, es el más habitual a utilizar en estos estudios, aparte de ser el más eficiente.

Para poder visualizar todo esto, al ya tener generado el emparejamiento anterior, utilizaremos la función *summary* para que nos muestre todo lo que aparece en la Figura 3.4. Esta nos mostrará la descriptiva generada de la base de datos original antes de emparejarse (*Summary of Balance for All Data*) y de la descriptiva de los individuos que están emparejados (*Summary of Balance for Matched Data*), así como los valores que se generan entre ambas bases de datos en cuanto a valores como el promedio o la desviación estándar y, finalmente, el número de individuos en cada grupo.

```
Call:
matchit(formula = treat ~ age + married + nodegree + re74 + re75,
    data = lalonde, method = "nearest", ratio = 1)

Summary of Balance for All Data:
          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
distance         0.3878        0.2640          0.9512     0.6735    0.2316   0.3485
age             25.8162       28.0303         -0.3094     0.4400    0.0813   0.1577
married          0.1892        0.5128         -0.8263          .    0.3236   0.3236
nodegree         0.7081        0.5967          0.2450          .    0.1114   0.1114
re74          2095.5737     5619.2365         -0.7211     0.5181    0.2248   0.4470
re75          1532.0553     2466.4844         -0.2903     0.9563    0.1342   0.2876


Summary of Balance for Matched Data:
          Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max Std. Pair Dist.
distance         0.3878        0.3793          0.0652     1.1053    0.0224   0.1892          0.0684
age             25.8162       25.6919          0.0174     0.3791    0.1055   0.2595          1.1279
married          0.1892        0.2162         -0.0690          .    0.0270   0.0270          0.2070
nodegree         0.7081        0.7135         -0.0119          .    0.0054   0.0054          0.5113
re74          2095.5737     2054.2118          0.0085     1.4983    0.0354   0.2270          0.4739
re75          1532.0553     1528.8677          0.0010     1.8413    0.0516   0.2216          0.6583

Percent Balance Improvement:
          Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
distance              93.1       74.7      90.3     45.7
age                   94.4      -18.2     -29.8    -64.5
married               91.6          .      91.6     91.6
nodegree              95.1          .      95.1     95.1
re74                  98.8       38.5      84.3     49.2
re75                  99.7    -1266.0      61.5     23.0

Sample Sizes:
            Control Treated
All             429     185
Matched         185     185
Unmatched       244       0
Discarded         0       0
```

Figura 3.4: Resultado de la función *summary* al modelo generado por PS *matching*.

En el resultado que obtenemos en el resumen de Lalonde[4] observamos que el promedio de las distancias, que equivale a los valores del PS, tiene un valor de 0.3878 en el grupo tratamiento y un valor de 0.2640 en el grupo control.

Cuando se hace el emparejamiento vemos, en el resumen de los datos emparejados, que estas diferencias se han suavizado para así hacer que no existan diferencias entre grupos y evitar una posible variable de confusión. En la tercera matriz observamos cómo han mejorado las variables al realizar el emparejamiento. En la última matriz vemos cómo se han emparejado los individuos, siendo todos los tratados emparejados pero muchos de los de control no lo han sido.

Para poder hacer el estudio a partir de aquí necesitaremos pasar los individuos emparejados a una nueva base de datos, donde encontramos la descriptiva univariante en el Cuadro 3.3. Esto lo

[4]Página 51 de apéndices

realizaremos con la función *match.data* que nos ordena los individuos que quedan en cada grupo generados con la función anterior. Genera una base de datos con las variables utilizadas en el modelo más 3 variables que consisten en el valor del PS calculado, denominado *distance*, *weights* que consiste en qué grupo de ponderación están, en este caso, solo existe un grupo, y *subclass* que te indica con que individuo esta emparejado, siendo este el qué coincida con el mismo valor de esta variable o tenga un valor similar.

treat	age	educ	married	nodegree	re74	re75	re78	distance	weights	subclass
0:185	Min. :16.00	Min. : 0.00	0:295	0:107	Min. : 0	Min. : 0.0	Min. : 0.00	Min. :0.01166	Min. :1	1 : 2
1:185	1st Qu.:18.00	1st Qu.: 9.00	1: 75	1:263	1st Qu.: 0	1st Qu.: 0.0	1st Qu.: 74.99	1st Qu.:0.28950	1st Qu.:1	2 : 2
	Median :23.00	Median :10.00			Median : 0	Median : 158.7	Median : 4084.50	Median :0.44803	Median :1	3 : 2
	Mean :25.75	Mean :10.08			Mean : 2075	Mean : 1530.5	Mean : 5996.64	Mean :0.38353	Mean :1	4 : 2
	3rd Qu.:29.00	3rd Qu.:12.00			3rd Qu.: 1869	3rd Qu.: 1890.1	3rd Qu.: 9144.17	3rd Qu.:0.46798	3rd Qu.:1	5 : 2
	Max. :55.00	Max. :18.00			Max. :35040	Max. :25142.2	Max. :60307.93	Max. :0.62932	Max. :1	6 : 2
										(Other):358

Cuadro 3.3: Resumen de la nueva base de datos generado con *match.data*.

A partir de aquí, antes de verificar si todo es correcto o no, podemos comprobar de diversas formas si hemos realizado bien el emparejamiento viendo algunas de las características de la base de datos formada ahora o con el modelo realizado. Para ver si el número de individuos está bien en la nueva base de datos y se han añadido bien todas las variables, podemos utilizar la función *dim* del paquete base de R [16] para ver las dimensiones de esta base de datos. También podemos realizar un *plot* sobre el modelo realizado para observar exactamente qué valores están o no emparejados y de qué grupo son estos. En la Figura 3.5 se pueden ver los individuos con valor de PS marcado en el eje horizontal y dividido en cuatro grupos, los tratados y no tratados de los individuos emparejados y los que finalmente no han sido emparejados.

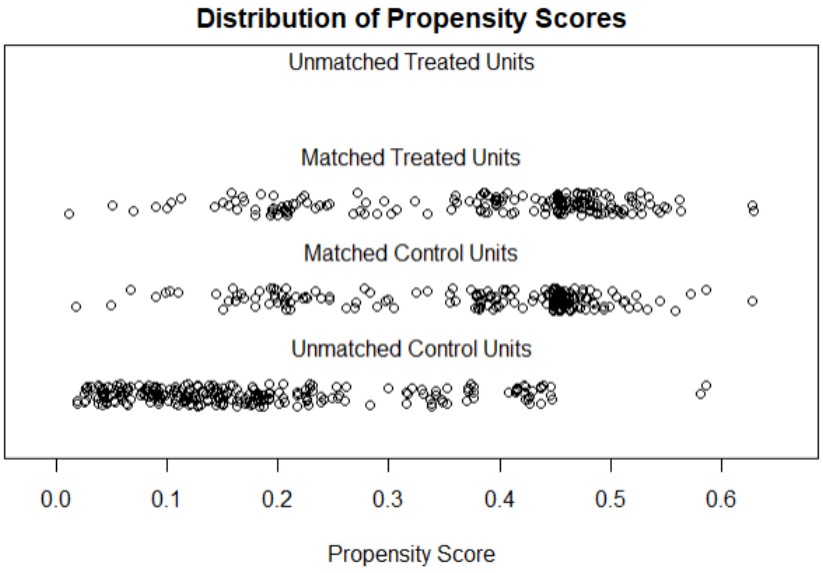
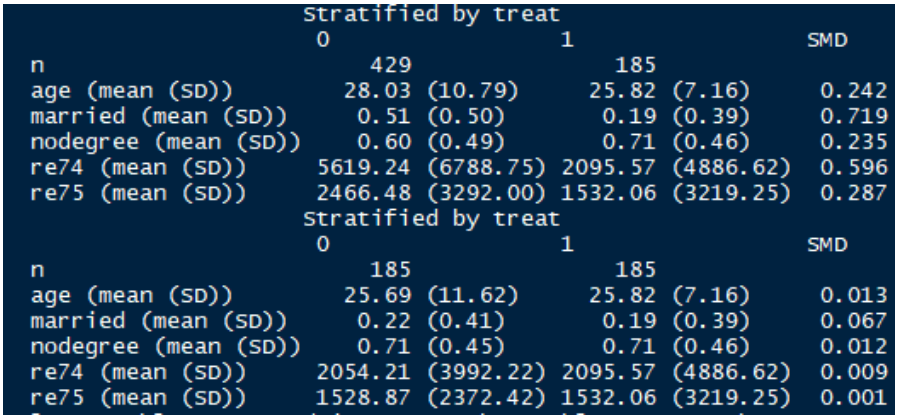


Figura 3.5: Los individuos de la muestra representados en cada grupo.

Cada punto visible en la Figura 3.5[5] representa un individuo con un cierto valor del PS en uno de los grupos a los que pertenece después de hacer el emparejamiento. Aquí vemos claramente como todos los individuos del grupo tratamiento (185) están emparejados y 244 del grupo control no lo están. Esto pasa debido a la gran diferencia entre el número de sujetos entre grupos.

Con la función *CreateTableOne*, podemos estudiar si aún existen variables de confusión o no en nuestro estudio.

[5]Página 54 de apéndices



```
                    Stratified by treat
                     0                1                SMD
n                      429              185
age (mean (SD))      28.03 (10.79)    25.82 (7.16)     0.242
married (mean (SD))   0.51 (0.50)      0.19 (0.39)     0.719
nodegree (mean (SD))  0.60 (0.49)      0.71 (0.46)     0.235
re74 (mean (SD))   5619.24 (6788.75) 2095.57 (4886.62) 0.596
re75 (mean (SD))   2466.48 (3292.00) 1532.06 (3219.25) 0.287
                    Stratified by treat
                     0                1                SMD
n                      185              185
age (mean (SD))      25.69 (11.62)    25.82 (7.16)     0.013
married (mean (SD))   0.22 (0.41)      0.19 (0.39)     0.067
nodegree (mean (SD))  0.71 (0.45)      0.71 (0.46)     0.012
re74 (mean (SD))   2054.21 (3992.22) 2095.57 (4886.62) 0.009
re75 (mean (SD))   1528.87 (2372.42) 1532.06 (3219.25) 0.001
```

Figura 3.6: Ejemplo de cómo se representa la función *CreateTableOne* en R con la base de datos Lalonde. Arriba antes de emparejamiento y abajo después de emparejamiento.

En la Figura 3.6[6] vemos como antes de aplicar este método existen diferencias entre los grupos de algunas de las variables. En los datos ya emparejados observamos como ya no quedan posibles variables de confusión en el estudio, ya que los valores SMD son todos menores a 0.1, como hemos mencionado en el apartado 3.2.

Teniendo hecho ya el estudio basado en el *propensity score* en nuestra base de datos y tratando correctamente nuestras variables de confusión, ya se podría empezar el análisis del efecto del tratamiento sobre la variable respuesta como si fuera un estudio aleatorizado. La base de datos correspondiente a este estudio será la generada por la función *match.data*, en la cual eliminaremos los individuos no emparejados juntada con la variable respuesta gracias a la función *merge*.

3.5. PS por ponderación

En este apartado ilustraremos cómo es el método IPW con PS en R. Para este estudio existe un paquete que es esencial, denominado *PSweight*. Este fue creado en 2020 por diversos integrantes de dos universidades de Estados Unidos, la Universidad de Yuke y Universidad de Duke en *PSweight: An R Package for propensity score Weighting Analysis* (2020) [17]. La mayor parte de este apartado

[6]Página 52 de apéndices

está basado en este articulo y el paquete que han creado. En este articulo mencionan diversas maneras de utilizar las ponderaciones pero nosotros solo utilizaremos las estimaciones más habituales que son las de IPW.

Lo primero que necesitaremos hacer es generar la fórmula del modelo para estudiar, es decir, saber que variables utilizar para analizar el resultado. En nuestro caso, utilizaremos la misma fórmula utilizada en el apartado anterior, que consiste en utilizar la variable *treat* como variable dependiente y las variables de confusión como variables independientes dentro del modelo realizado.

Los diferentes estratos y valores del PS como los puntajes de balance los generaremos con una función llamada *SumStat*. Si hacemos un *summary* junto a esta función obtendremos un resumen de las variables de la base de datos por todos los individuos, por el grupo tratamiento y por el grupo control, muy parecida a la que se genera con la función *CreateTableOne*, ya que también muestra el SMD.

```
weights estimated for:  IPW
unweighted result
           Mean 0   Mean 1   SMD
age        28.030   25.816 0.242
married     0.513    0.189 0.719
nodegree    0.597    0.708 0.235
re74     5619.237 2095.574 0.596
re75     2466.484 1532.055 0.287

IPW result
           Mean 0   Mean 1   SMD
age        27.301   26.089 0.135
married     0.415    0.446 0.069
nodegree    0.630    0.557 0.155
re74     4588.482 7782.340 0.540
re75     2198.306 3371.090 0.360
```

Figura 3.7: Resumen de la ponderación IPW estratificado con la variable *treat*.

En la Figura 3.7[7] vemos como quedarían los grupos después de realizar las ponderaciones. Aquí vemos como con este método no se corrigen bien las posibles variables de confusión, ya que los valores de SMD son grandes y, por consecuente, vemos como las diferencias aun son mayores. Aun así, seguiremos con el estudio para ilustrar como se realiza este método.

[7]Página 53 de apéndices

Con esto hecho, podemos realizar diferentes gráficos para poder ver como se expresan los resultados visualmente. Si utilizamos la función *plot* junto al complemento *density*, generaremos un gráfico de densidad y podremos observar como se distribuyen los valores del PS para los dos grupos de tratamiento en los dos grupos generados por ponderación. También podemos representar estos valores con un histograma con la función *plot* junto a *hist*. Estos gráficos son los mismos que hemos visto en el apartado 3.3 debido a que los valores del PS son los mismos. Finalmente, podemos representar los valores SMD de cada tipo de estudio por ponderación del puntaje de balance con la función *plot* junto a *balance*.

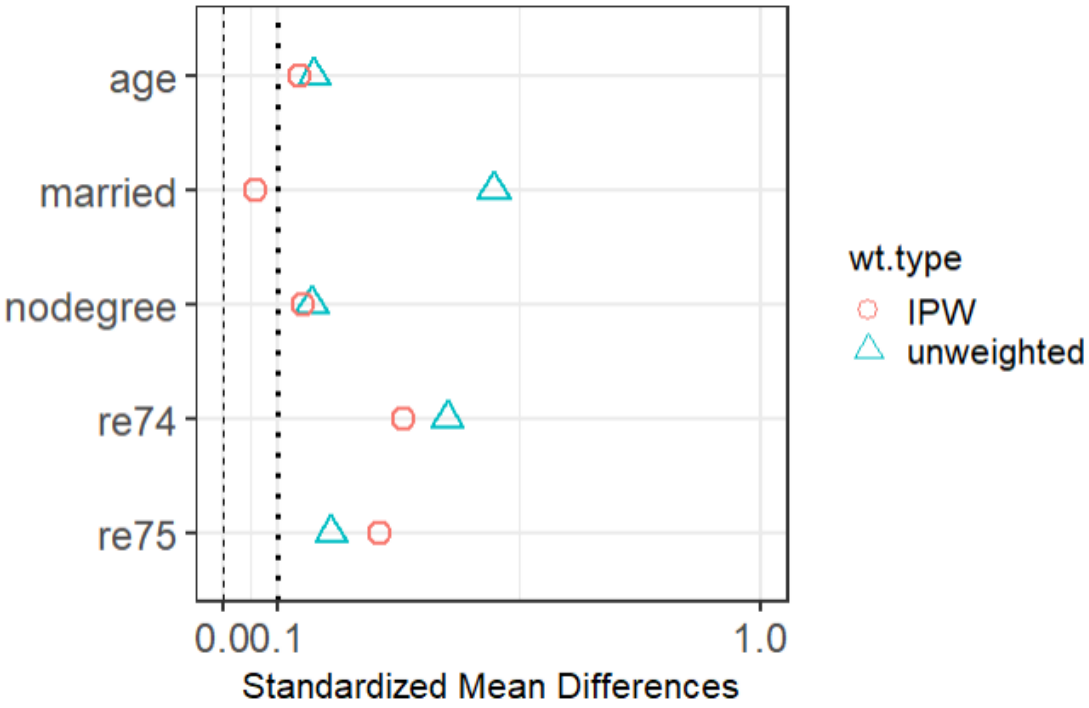


Figura 3.8: Comparación del SMD en los datos de Lalonde antes y después de ponderar.

En la Figura 3.8[8] observamos como son los valores de SMD para cada variable antes (*unweighted*) y después de ponderar (*IPW*). Vemos como solo se corrige la variable *married*, por consecuente, y como hemos visto antes, con estos datos no es recomendable utilizar el método por

[8]Página 54 de apéndices

ponderación.

Para finalizar, añadiremos los valores de PS generados por la función *SumStat* a la base de datos con los datos generados en *propensity* y la primera columna de este *dataframe*.

3.6. PS por estratificación

En este apartado ilustraremos cómo es el método de estratificación con PS en R. Para utilizar este método utilizaremos principalmente el paquete *PSAgraphics*, el cual esta explicado en detalle en el articulo de *PSAgraphics: An R Package to Support propensity score Analysis* (2009) [18].

Lo primero que necesitaremos hacer es un modelo de regresión logística utilizando la misma fórmula que los métodos anteriores. Utilizaremos la función *summary* para ver los coeficientes calculados para este modelo. Recordemos que para calcular el valor de PS de un individuo, necesitaremos utilizar estos coeficientes beta generados en el modelo y los valores de cada individuo para poder calcularlo. Esto lo podemos calcular automáticamente gracias a que dentro del modelo ya se han generado estos valores. Estos se pueden recoger utilizando los valores *fitted* dentro del modelo.

A partir de aquí, decidiremos en cuántos estratos queremos separar a nuestros individuos y cómo estudiarlo. Utilizaremos 5 estratos para ilustrar este método en R. Para poder generar estos estratos, podemos dividir nuestros datos con la función *cut*, que nos divide en los trozos que nosotros queramos un vector seleccionado. En nuestro caso, dividiremos el vector generado con los valores *fitted* del modelo, que son los mismos que los valores del PS. Con esta función obtendremos otro vector que nos indicará, para cada valor del PS, a qué estrato pertenece.

A continuación, analizaremos estos grupos con la función *box.psa*, la cual nos comparará cada grupo de estrato con los diferentes grupos de tratamiento para las diversas variables de confusión que sean continuas, comparando las medias de estos 2 grupos teniendo en cuenta una variable independiente. Esto lo tendremos que realizar para todas las variables de confusión que sean continuas que tengamos en nuestro estudio.

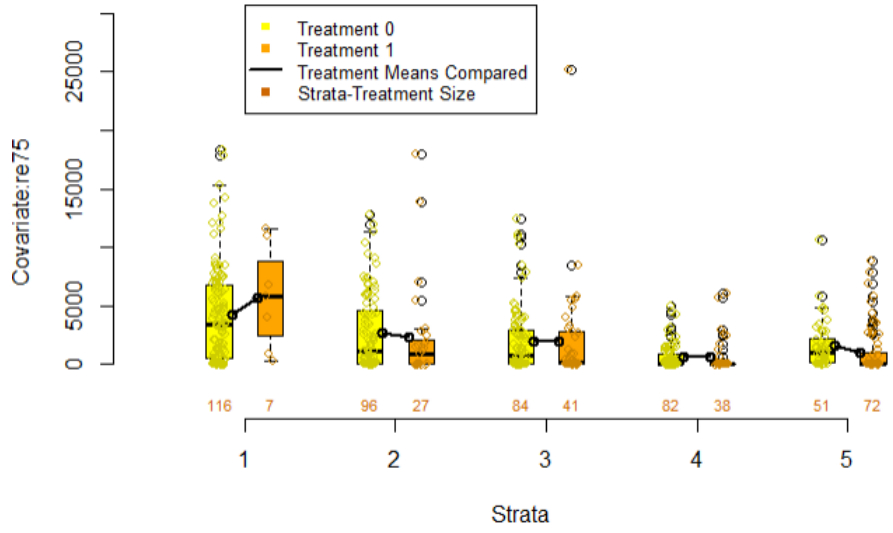


Figura 3.9: Estratificación para comparar la variable *re75*.

En la Figura 3.10[9] podemos ver que diferencias hay entre grupos para la variable *re75*. Vemos como en el segmento dividido en 5 estratos, en el primero es el único donde podría existir una diferencia entre los dos grupos si hablamos del promedio de los valores obtenidos. En los otros no parece existir una diferencia significativa.

También se puede realizar este estudio con una variable no continua si utilizamos la función *cat.psa*. En este caso utilizaremos para las variables *nodegree* y *married*. No es recomendable realizarlo para variables con muchos valores diferentes, ya que en estos gráficos no podremos analizar muy bien las diferencias debido al bajo número de individuos con valores específicos. En el caso de que la variable *educ*, que posee diversos valores no continuos, fuese una variable de confusión necesitaríamos realizar esto para más de un valor y observaríamos lo difícil que puede ser estudiarlo de este modo.

[9]Página 55 de apéndices

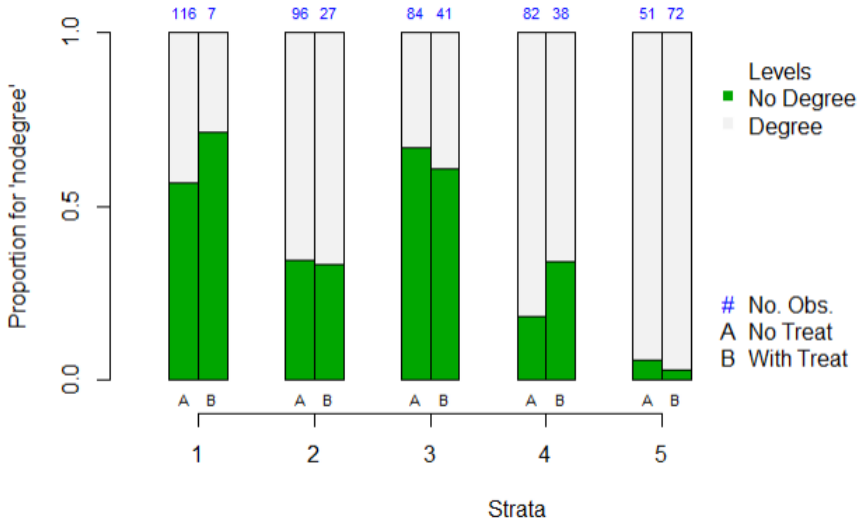


Figura 3.10: Estratificación para comparar la variable *nodegree*.

En la Figura 3.11[10] hemos realizado el estudio para la variable binaria *nodegree*. En estas gráficas podemos ver el promedio de valores de esta variable sobre cada estrato, al igual que la anterior. Pero esta función, *cat.psa*, es solo para variables no continuas. Podemos observar como el primer y cuarto estrato podrían existir diferencias entre los grupos.

Deberíamos realizar este proceso para todas las variables de confusión. Al estudiar todas ellas vemos como debemos eliminar el primer estrato y el cuarto porque estos pueden contener alguna variable de confusión.

Ya para finalizar este método, para añadir estos valores a la base de datos teniéndolos ya analizados solo hemos de añadir los valores del PS estratificados con la variable *cut* a nuestra base de datos ya existente.

[10]Página 56 de apéndices

3.7. PS por regresión de covariables

En este apartado ilustraremos cómo se implemente el método de regresión de covariables con PS. Bajo este método, se comienza calculando los valores PS mediante un modelo de regresión utilizando la variable *treat* como variable dependiente y las variables de confusión como variables independientes; los valores PS son los valores *fitted* del modelo. Teniendo esto, lo único que tenemos que realizar es añadir estos valores a la base de datos como si fuese una covariable más. Luego, podríamos evaluar el efecto del tratamiento sobre la variable respuesta, pero en este caso, utilizando solo la variable tratamiento y la variable PS como variables independientes. Esto permitiría estimar el efecto del tratamiento sobre la variable respuesta.

Esto lo podemos estudiar utilizaremos la función *summary* en el modelo para comprobar esto. Cuando tengamos este modelo con una variable respuesta que tiene una distribución binomial, podemos utilizar también la función *logistic.display* del paquete *epiDisplay* [19] para ver si las variables tratamiento ó PS son significativas en el modelo de regresión logística.

```
Call:
lm(formula = re78 ~ treat + ps, data = newdata)

Residuals:
   Min     1Q Median     3Q    Max
 -9903  -5351  -1863   3859  55339

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9935.5      624.8  15.901  < 2e-16 ***
treat          748.3      685.0   1.093    0.275
ps          -11178.7     1956.0  -5.715 1.72e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7285 on 611 degrees of freedom
Multiple R-squared:  0.05219,   Adjusted R-squared:  0.04909
F-statistic: 16.82 on 2 and 611 DF,  p-value: 7.732e-08
```

Figura 3.11: Modelo lineal para ver el efecto de *treat* y el PS.

En la Figura 3.11[11] vemos el modelo lineal para ver el efecto del tratamiento junto a los valores del PS. Vemos como el efecto del tratamiento no es estadísticamente significativo con respecto a

[11] Página 57 de apéndices

la variable respuesta *re78*. Observamos asimismo, que la variable PS es significativa y que hicimos bien en ajustar por ella.

En la Figura 3.12 reportamos el modelo lineal utilizando solo la variable tratamiento como variable independiente y la variable respuesta como dependiente.

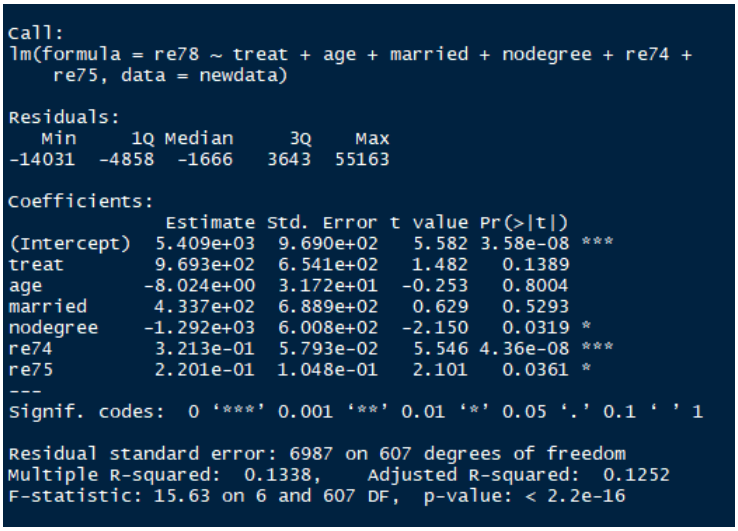
```
Call:
lm(formula = re78 ~ treat, data = newdata)

Residuals:
   Min     1Q Median     3Q    Max
 -6984  -6349  -2048   4100  53959

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6984.2      360.7  19.362   <2e-16 ***
treat         -635.0      657.1  -0.966    0.334
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7471 on 612 degrees of freedom
Multiple R-squared:  0.001524,  Adjusted R-squared:  -0.0001079
F-statistic: 0.9338 on 1 and 612 DF,  p-value: 0.3342
```

Figura 3.12: Modelo lineal para ver el efecto de *treat*.

En este modelo[12] vemos también que el tratamiento no afecta significativamente al hecho de tener más o menos salario anual.

Finalmente, en la Figura 3.13 veremos cuál sería el efecto de tratamiento ajustando, de forma clásica, por todas las covariables de confusión:

[12] Página 57 de apéndices



```
Call:
lm(formula = re78 ~ treat + age + married + nodegree + re74 +
    re75, data = newdata)

Residuals:
   Min     1Q Median     3Q    Max
-14031  -4858  -1666   3643  55163

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.409e+03  9.690e+02   5.582 3.58e-08 ***
treat        9.693e+02  6.541e+02   1.482   0.1389
age         -8.024e+00  3.172e+01  -0.253   0.8004
married      4.337e+02  6.889e+02   0.629   0.5293
nodegree    -1.292e+03  6.008e+02  -2.150   0.0319 *
re74         3.213e-01  5.793e-02   5.546 4.36e-08 ***
re75         2.201e-01  1.048e-01   2.101   0.0361 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6987 on 607 degrees of freedom
Multiple R-squared:  0.1338,    Adjusted R-squared:  0.1252
F-statistic: 15.63 on 6 and 607 DF,  p-value: < 2.2e-16
```

Figura 3.13: Modelo lineal para ver el efecto de *treat* ajustado por variables de confusión.

En este modelo[13] podemos ver como el efecto del tratamiento no es significativo frente a la variable respuesta. En cambio, las variables que son significativas son las *nodegree*, *re74* y *re75*. Tiene sentido que estas sean significativas debido a priori son las más relacionadas con el salario de una persona.

La conclusión que sacamos de los 3 modelos es que no podemos afirmar que haya un efecto del tratamiento respecto a si un paciente puede tener un salario más alto o no.

En este capítulo hemos ilustrado como implementar en R los 4 métodos basados en el uso del PS y hemos comentado los resultados pertinentes. Queremos resaltar que la realización de un modelo final no era objetivo de este trabajo y no desarrollamos aquí ese análisis más exhaustivo.

[13] Página 57 de apéndices

Capítulo 4

ANÁLISIS CON METODOLOGÍA PS: DATOS NHANES 2018

4.1. Descripción de la base de datos NHANES 2018

En este capítulo aplicaremos todo lo aprendido anteriormente en una base de datos reciente. El estudio donde nos basaremos es de datos extraídos del *National Center for Health Statistics* (NHANES). El objetivo del análisis es determinar si pacientes con artritis, variable tratamiento *arthritis.type*, están en mayor riesgo de sufrir un ataque al corazón, variable respuesta *heart.attack*[1].

Todas las variables de este estudio son no continuas como variables independientes. A continuación describiremos todas ellas:

- *heart.attack*: **variable respuesta** de nuestro estudio. Consiste en si el individuo ha tenido alguna vez un ataque al corazón recientemente. Hay 3439 individuos que no han padecido un ataque al corazón 110 que si.

- *arthritis.type*: **variable tratamiento** del estudio. Consiste en ver que individuo tiene artritis

[1]Todo el estudio esta realizado con software R, como en el capítulo anterior, y se podrá ver todo el código en los apéndices. Para cargar la base de datos, el código está en la página 60.

o no. Hay 666 individuos tienen artritis y 2883 individuos no tienen artritis. Tomaremos a los individuos que tienen artritis como grupo tratamiento y a los que no como grupo control.

- *gender*: variable que indica el sexo del individuo. En esta base de datos hay 1734 hombres y 1815 mujeres.

- *bmi*: variable que indica el bmi del individuo, evaluado por intervalos. En el intervalo de 0 a 25 hay 1004 individuos y de 25 a 80 hay 2545.

- *diabetes*: variable que indica si el individuo tiene diabetes o no. Hay 3124 que no tienen diabetes y 425 si tienen.

- *smoke*: variable que indica si el individuo fuma o no. Hay 2114 que no son fumadores y 1435 fuman.

- *age*: variable que indica la edad del individuo, por intervalos. De los 20 a 50 años hay 1857 individuos, entre 50 y 70 años hay 1197 individuos y mayores de 70 años hay 495.

- *marriage*: variable que indica si el individuo se ha casado o no. Hay 1354 que no están casados y 2195 están casados.

- *annualincome*: variable que indica el el salario anual de un individuo en dolares. Hay 521 que cobran menos de 20000 dolares anuales, 1385 que cobran entre 20000 y 49999 dolares anuales y 1643 que cobran más de 50000 dolares anuales.

- *physical.activity*: variable que indica si el individuo realiza actividad física y si esta es moderada o hace mucho deporte. Hay 1841 individuos que no hacen deporte, 897 que hacen mucho deporte y 811 que hacen deporte pero de forma moderada.

- *medical.access*: variable que indica si el individuo tiene acceso a sanidad. Esta base de datos esta basada en estados unidos y recordemos que ahí la sanidad es de acceso privado. Hay 701 individuos que no tienen acceso a sanidad y 2848 que si lo tienen.

- *blood.pressure*: variable que indica si el individuo ha tenido en algún momento de su vida una presión arterial elevada. Hay 2362 individuos que si la han tenido y 1187 que no la han tenido.

- *healthy.diet*: variable que indica si el individuo tiene actualmente una buena dieta o no. Hay

229 que tienen una mala dieta, 920 tienen que una dieta aceptable y 2400 que tienen una buena dieta.

heart.attack	arthritis.type	gender	bmi	diabetes	smoke	age
No :3439	Non-arthritis:2883	Male :1734	(0,25] :1004	No :3124	No :2114	(19.9,50]:1857
Yes: 110	Yes-arthritis: 666	Female:1815	(25,80]:2545	Yes: 425	Yes:1435	(50,70] :1197
						(70,81] : 495

marriage	annualincome	physical.activity	medical.access	blood.pressure	healthy.diet
Not.married:1354	$<$20k : 521	No :1841	No : 701	No :2362	Poor: 229
Married :2195	20kto54k:1385	High : 897	Yes:2848	Yes:1187	Fair: 920
	55k+ :1643	Moderate: 811			Good:2400

Cuadro 4.1: Descriptiva univariante de los datos NHANES 2018.

4.2. Detección de variables de confusión

Como hemos hecho en el capítulo 3, estudiaremos los valores promedio de cada variable para cada estrato de la variable tratamiento con la función *CreateTableOne* y veremos que variables independientes afectan a la variable respuesta con *compareGroups*:

	Stratified by arthritis.type		
	Non-arthritis	Yes-arthritis	SMD
n	2883	666	
gender = Female (%)	1403 (48.7)	412 (61.9)	0.268
bmi = (25,80] (%)	2018 (70.0)	527 (79.1)	0.211
diabetes = Yes (%)	315 (10.9)	110 (16.5)	0.163
smoke = Yes (%)	1098 (38.1)	337 (50.6)	0.254
age (%)			0.978
(19.9,50]	1734 (60.1)	123 (18.5)	
(50,70]	866 (30.0)	331 (49.7)	
(70,81]	283 (9.8)	212 (31.8)	
marriage = Married (%)	1803 (62.5)	392 (58.9)	0.075
annualincome (%)			0.226
<20k	383 (13.3)	138 (20.7)	
20kto54k	1118 (38.8)	267 (40.1)	
55k+	1382 (47.9)	261 (39.2)	
physical.activity (%)			0.134
No	1493 (51.8)	348 (52.3)	
High	755 (26.2)	142 (21.3)	
Moderate	635 (22.0)	176 (26.4)	
medical.access = Yes (%)	2233 (77.5)	615 (92.3)	0.425
blood.pressure = Yes (%)	794 (27.5)	393 (59.0)	0.670
healthy.diet (%)			0.054
Poor	182 (6.3)	47 (7.1)	
Fair	738 (25.6)	182 (27.3)	
Good	1963 (68.1)	437 (65.6)	

Figura 4.1: Descriptiva estratificada de la variable *arthritis.type* **antes** de realizar el emparejamiento.

	No N=3439	Yes N=110	p.overall
arthritis.type:			0.001
Non-arthritis	2807 (81.6%)	76 (69.1%)	
Yes-arthritis	632 (18.4%)	34 (30.9%)	
gender:			<0.001
Male	1651 (48.0%)	83 (75.5%)	
Female	1788 (52.0%)	27 (24.5%)	
bmi:			0.573
(0,25]	976 (28.4%)	28 (25.5%)	
(25,80]	2463 (71.6%)	82 (74.5%)	
diabetes:			<0.001
No	3048 (88.6%)	76 (69.1%)	
Yes	391 (11.4%)	34 (30.9%)	
smoke:			<0.001
No	2069 (60.2%)	45 (40.9%)	
Yes	1370 (39.8%)	65 (59.1%)	
age:			<0.001
(19.9,50]	1848 (53.7%)	9 (8.18%)	
(50,70]	1145 (33.3%)	52 (47.3%)	
(70,81]	446 (13.0%)	49 (44.5%)	
marriage:			0.270
Not.married	1306 (38.0%)	48 (43.6%)	
Married	2133 (62.0%)	62 (56.4%)	
annualincome:			<0.001
<20k	485 (14.1%)	36 (32.7%)	
20kto54k	1338 (38.9%)	47 (42.7%)	
55k+	1616 (47.0%)	27 (24.5%)	
physical.activity:			0.394
No	1777 (51.7%)	64 (58.2%)	
High	872 (25.4%)	25 (22.7%)	
Moderate	790 (23.0%)	21 (19.1%)	
medical.access:			0.006
No	691 (20.1%)	10 (9.09%)	
Yes	2748 (79.9%)	100 (90.9%)	
blood.pressure:			<0.001
No	2329 (67.7%)	33 (30.0%)	
Yes	1110 (32.3%)	77 (70.0%)	
healthy.diet:			0.001
Poor	214 (6.22%)	15 (13.6%)	
Fair	903 (26.3%)	17 (15.5%)	
Good	2322 (67.5%)	78 (70.9%)	

Figura 4.2: Descriptiva estratificada de la variable *heart.attack*.

Al comparar los dos grupos[2] de individuos estratificando con la variable tratamiento, Figura 4.1, vemos como en la mayoría de las variables hay una diferencia grande en el valor promedio de los datos, excepto *marriage* y *healthy.diet*. En la Figura 4.2 vemos como las variables que no afectan a la variable respuesta son *bmi*, *marriage*, *physical.activity*. Tomaremos como variables de confusión aquellas que tienen un SMD mayor a 0.1 en la Figura 4.1 y un *p.overall* inferior a 0.05 en la Figura 4.2. En este caso vemos como las variables de confusión son las siguientes: *gender*, *diabetes*, *smoke*, *age*, *annualincome*, *medical.access* y *blood.pressure*.

4.3. Análisis de los datos NHANES con metodología PS

4.3.1. Elección de método

Para realizar el análisis por PS debemos pensar que método es mejor para nuestro estudio. En el capítulo 3 hemos implementado todos los métodos pero, por lo general, no es útil utilizar todos sino pensar cuál nos viene mejor.

Lo primero que vemos es que hay mucha diferencia entre el número de individuos que hay entre el grupo tratamiento y el grupo control. Por eso lo mejor sería utilizar directamente un PS por emparejamiento, porque uno de los motivos por los que pueden existir nuestras variables de confusión es debido a este desnivel entre el número de individuos en cada grupos.

Vamos a realizar también el método de regresión por covariables para comparar el modelo que se realizará con PS y el modelo lineal generalizado de la base de datos original.

Se podría utilizar el método por estratos, pero si tenemos que escoger entre emparejamiento y estratificación, siempre escogeremos el método de emparejamiento, debido a que es más práctico y eficaz.

Por el mismo motivo por el que aceptamos hacer el método por emparejamiento, descartaríamos directamente utilizar los métodos por ponderación. Necesitamos que los dos estratos tengan un número equitativo de individuos y, en el momento que es posible utilizar el método por empareja-

[2]Página 68-69 de apéndices

miento, siempre será mejor priorizar utilizar este último.

4.3.2. Estudio del PS en la base de datos

PS *MATCHING*

Al haber decidido que realizaríamos el estudio por emparejamiento pasaremos la base de datos por prácticamente el mismo código que en el capítulo 3. Como hemos explicado en el apartado 3.4 realizaremos el estudio mayormente con el paquete *MatchIt*. Empezamos con el emparejamiento con la función *matchit*. Generamos la nueva base de datos con la función *match.data*. Realizamos la descriptiva estratificada[3] para la variable tratamiento para ver si se han balanceado las variables de confusión:

	Stratified by arthritis.type		
	Non-arthritis	Yes-arthritis	SMD
n	666	666	
gender = Female (%)	387 (58.1)	412 (61.9)	0.077
diabetes = Yes (%)	115 (17.3)	110 (16.5)	0.020
smoke = Yes (%)	356 (53.5)	337 (50.6)	0.057
age (%)			0.091
(19.9,50]	122 (18.3)	123 (18.5)	
(50,70]	358 (53.8)	331 (49.7)	
(70,81]	186 (27.9)	212 (31.8)	
annualincome (%)			0.039
<20k	131 (19.7)	138 (20.7)	
20kto54k	279 (41.9)	267 (40.1)	
55k+	256 (38.4)	261 (39.2)	
medical.access = Yes (%)	619 (92.9)	615 (92.3)	0.023
blood.pressure = Yes (%)	384 (57.7)	393 (59.0)	0.027

Figura 4.3: Descriptiva estratificada de la variable *arthritis.type* **después** de realizar el emparejamiento.

Verificación de balance después del emparejamiento

Lo primero que vemos en este método es que se han emparejado todos los 666 individuos del grupo control de la variable tratamiento pero hemos perdido 2217 individuos, que representa aproximadamente tres cuartos de los individuos de este grupo.

[3] Página 72 de apéndices

Para verificar que el balance está bien realizado, observaremos los valores de la diferencia entre valores promedio estandardizado en la Figura 4.3. Si lo comparamos con el los valores SMD originales, en la Figura 4.1, vemos como estos datos mejoran mucho el estudio debido a que eliminan los efectos de las variables de confusión. En el caso de *gender* vemos como su SMD pasa de 0.268 a 0.77. Con la variable *diabetes* pasa de 0.163 a 0.020. En el caso de la variable *smoke* observamos como pasa de 0.254 a 0.57. La variable *age* originalmente el SMD era de 0.978 y despues del emparejamiento es de 0.091. En la variable *annualincome* hemos pasado de un valor SMD de 0.226 a 0.039. En el caso de la variable *medical.access* pasa de 0.425 a 0.023. Finalmente, la variable *blood.pressure* pasa de 0.67 a 0.027.

En conclusión, es bueno utilizar PS *matching* porque elimina todas las variables de confusión porque todos los valores SMD de cada variable pasan a ser inferiores a un umbral, 0.1.

Resultados antes y después de emparejar

Como la variable respuesta *heart.attack* es binaria, a diferencia de la de Lalonde que era continua, podemos utilizar la función *logistic.display* del paquete *epiDisplay* [19] para ver los *odds ratio* de las variables independientes y los p-valores de cada test. En este apartado nos centraremos más en los *odds ratio* (OR). Este valor corresponde a elevar a la exponencial el valor obtenido de un coeficiente en una variable independiente en un modelo.

$$OR_i = e^{B_i} \tag{4.1}$$

Donde B_i corresponde al coeficiente extraído del modelo de la variable i y OR_i es el valor del *odds ratio* de la variable i.

Primero realizaremos el análisis para los datos sin emparejar:

```
Logistic regression predicting heart.attack : Yes vs No

                                          OR(95%CI)       P(Wald's test) P(LR-test)
arthritis.type: Yes-arthritis vs Non-arthritis 1.99 (1.31,3)  0.001          0.002

Log-likelihood = -485.5788
No. of observations = 3549
AIC value = 975.1575
```

Figura 4.4: Resultado de la función *logistic.display* para la función con solo la variable tratamiento **antes** del emparejamiento.

En la figura 4.4 observamos como, antes de realizar el emparejamiento por PS, nos encontramos con que el valor de la *odds ratio* de la variable tratamiento frente a la variable respuesta es de 1.99. Este indica que pacientes con artritis tienen una mayor probabilidad de sufrir un ataque al corazón comparado con los que no tienen artritis. Vemos como el efecto la variable tratamiento sobre la variable respuesta es significativo. Sabemos que esto puede cambiar porque, en el apartado 4.2 hemos detectado presencia de variables de confusión.

Vamos a hacer lo mismo que en el modelo anterior pero con los datos generados con el emparejamiento:

```
Logistic regression predicting heart.attack : Yes vs No

                                          OR(95%CI)         P(Wald's test) P(LR-test)
arthritis.type: Yes-arthritis vs Non-arthritis 0.82 (0.51,1.31)  0.41           0.409

Log-likelihood = -288.3357
No. of observations = 1333
AIC value = 580.6715
```

Figura 4.5: Resultado de la función *logistic.display* para la función con solo la variable tratamiento **después** del emparejamiento.

En la Figura 4.5 vemos como, después de realizar el emparejamiento[4], nos queda que el efecto de la variable tratamiento no es significativo sobre la variable respuesta. Vemos como el valor de las *odds* después de emparejar nos encontramos que es de 0.82. Este indica que pacientes sin artritis tienen una mayor probabilidad de sufrir un ataque al corazón comparado con los que si tienen

[4]Página 73 de apéndices

artritis. Esto difiere completamente con lo que hemos obtenido en el primero modelo, siendo que era mucho más probable de que tengas un ataque al corazón teniendo artritis.

PS POR REGRESIÓN DE COVARIABLES

Como hemos explicado en el apartado 3.7, para realizar este método necesitamos calcular el PS mediante regresión logística ajustando la variable tratamiento *arthritis.type versus* las variables de confusión. Una vez calculados los valores del PS para cada individuo, añadiremos estos valores a la base de datos original. En este punto ya podemos proceder a estimar el efecto del tratamiento sobre la variable respuesta considerando solo la variable tratamiento y PS como variables independientes.

Dado que la respuesta es dicotómica, ajustaremos un modelo de regresión logística para ver si la variable tratamiento *arthritis.type* y la variable PS afectan a la variable respuesta *heart.attack*.

```
Logistic regression predicting heart.attack : Yes vs No

                                          crude OR(95%CI)      adj. OR(95%CI)       P(Wald's test) P(LR-test)
arthritis.type: Yes-arthritis vs Non-arthritis 1.99 (1.31,3)       0.89 (0.56,1.42)     0.627          0.626

ps (cont. var.)                           65.1 (24.96,169.78)  72.79 (25.29,209.51) < 0.001        < 0.001

Log-likelihood = -455.6817
No. of observations = 3549
AIC value = 917.3634
```

Figura 4.6: Resultado de la función *logistic.display* con el modelo *heart.attack* $\sim$ *arthritis.type + ps*.

En este modelo[5] vemos como tener o no artritis no es significativa a la hora de tener un ataque al corazón aun teniendo el mismo *odds ratio* que en la Figura 4.4. Lo que si observamos es que el efecto de la variable PS afecta significativamente a la variable respuesta. Por esto, ha sido buena elección utilizar el PS en estos datos.

[5]Página 73 de apéndices

Modelo con todas las covariables de confusión

A continuación, en la Figura 4. ajustaremos el modelo de regresión logística ”clásico” para la variable respuesta pero ajustando por todas las variables de confusión. Así podemos ver si son similares o distintos a los obtenidos ajustando por el PS

```
Call:
glm(formula = heart.attack ~ arthritis.type + gender + diabetes +
    smoke + age + annualincome + medical.access + blood.pressure,
    family = binomial(), data = DT9a)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9638  -0.2527  -0.1267  -0.0747   3.5590

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                -4.59701    0.46988  -9.783  < 2e-16 ***
arthritis.typeYes-arthritis -0.05112    0.23069  -0.222 0.824638
genderFemale               -1.16538    0.24450  -4.766 1.87e-06 ***
diabetesYes                 0.34846    0.23174   1.504 0.132661
smokeYes                    0.11932    0.21526   0.554 0.579375
age(50,70]                  1.76265    0.37762   4.668 3.05e-06 ***
age(70,81]                  2.41192    0.39297   6.138 8.38e-10 ***
annualincome20kto54k       -0.68853    0.23984  -2.871 0.004094 **
annualincome55k+           -1.28209    0.27393  -4.680 2.86e-06 ***
medical.accessYes           0.34509    0.35748   0.965 0.334379
blood.pressureYes           0.84650    0.23329   3.628 0.000285 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 980.82  on 3548  degrees of freedom
Residual deviance: 787.16  on 3538  degrees of freedom
AIC: 809.16

Number of Fisher Scoring iterations: 8
```

Figura 4.7: Modelo con todas las covariables de confusión y la variable tratamiento **antes** de aplicar el PS.

En este modelo[6] vemos como hay muchas variables que afectan significativamente al hecho de tener un ataque al corazón o no. En este caso, el efecto de la variable tratamiento no es significativo.

Conclusión del uso de metodología PS para datos NHANES

Observamos que al utilizar el método de regresión por covariables, los valores del PS afectan significativamente al modelo y, por consecuente, las variables de confusión afectan a como vemos el efecto del tratamiento sobre la variable respuesta.

[6]Página 74 de apéndices

Capítulo 5

CONCLUSIONES, DISCUSIÓN Y LIMITACIONES

El principal objetivo de estudiar el *propensity score* en este trabajo final de grado ha sido el de ilustrar como su uso, para estimar el efecto de un tratamiento en los estudios observacionales, conlleva a reducir el sesgo de confusión. Hemos visto que el uso de cualquiera de los 4 métodos basado en el PS nos ha permitido tratar un estudio observacional como si fuese uno aleatorizado (RCT); es decir, nos permite realizar un análisis estadístico final donde el efecto del tratamiento sobre la respuesta de interés se deba al tratamiento y no a las otras covariables de confusión.

En este trabajo hemos explicado como calcular el PS mediante regresión logística y cómo implementar en el software estadístico R los 4 métodos basados en su uso emulando el diseño de estudios aleatorizados. Además, se han usado herramientas estadísticas para verificar el balance de distribución de covariables. Toda esta metodología se ha ilustrado con datos reales: su cálculo e implementación (en el capítulo 3) con una base de datos más antigua y luego con una base de datos más reciente (capítulo 4). Todo el código R usado se incluye en los apéndices A y B.

Al realizar el estudio por PS debemos tener en cuenta que en algunos métodos, emparejamiento y estratificación, perdemos información con la eliminación de individuos del estudio; esto podría producir estimaciones desviadas de la realidad. Por ejemplo, en el estudio de la base de datos

NHANES vemos como al realizar el emparejamiento perdemos más de 2000 individuos del grupo control.

Por lo tanto, utilizar estos métodos basados en el uso del PS hace que tengamos una nueva base de datos mucho más reducida que la original y la pérdida de información podría ser muy grande. Por otra parte, la metodología PS nos permite analizar datos observacionales como si fuesen aleatorizados y así evitar un sesgo de confusión; hemos de valorar los ventajas y desventajas en cada caso. También nos podemos encontrar con la situación de tener muchas variables en el estudio. Por consiguiente, podría ser complicado eliminar todas las posibles variables de confusión debido a que en el momento en el que balanceamos una variable puede que estemos desbalanceando otra.

El uso de un método PS sobre otro requiere del conocimiento experto sobre ellos, pero también de las características de nuestras bases de datos como, por ejemplo, su tamaño muestral y número de covariables. Esto, sin embargo, sucede con cualquier otra metodología estadística: haya que valorar siempre sus ventajas y desventajas.

Para concluir, mencionar que los dos métodos más comúnmente usados en la investigación biomédica son el PS matching y el de regresión de covariables ajustando por el PS; los 2 métodos utilizados en el capítulo 4 para estimar el efecto del tratamiento con la base de datos NHANES. Hemos mostrado que ambos métodos basados en el PS mitigan el sesgo de confusión.

Limitaciones

Durante la realización de este trabajo de fin de grado nos hemos encontrado con una limitación: en un principio se iba a estudiar una base de datos de Cataluña que hubiese hecho más interesante el estudio, pero no obtuvimos el consentimiento para utilizarla. Para solucionar este problema se ha utilizado una base de datos de *open data* sobre datos de salud en Estados Unidos de la organización *Natiuonal Center for Health Statistics*. En concreto, de la sección *National Health And Nutrition Examinatión Survey*. Estos datos los hemos extraído con el software R, como se puede observar en el apéndice 2.

El objetivo de este trabajo final de grado era el estudio e implementación de metodología PS con estudios observacionales, lo cual hemos logrado pensando más en el usuario aplicado. Sería, sin embargo, deseable un análisis metodológico más exhaustivo.

Bibliografía

[1] Paul R. Rosenbaum, Donald B. Rubin (1983) *The central role of the propensity score in observational studies for causal effects* Biometrika, 70(1), 41-55

[2] Peter C. Austin (2011) *An introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies* variate Behavioral Research, 46:399–424

[3] Imbens, G. W. (2004). *Nonparametric estimation of average treatment effects under exogeneity: A review.* The Review of Economics and Statistics, 86, 4–29.

[4] Núria Correa Mañas, Jesús Cerquides, Joan Capdevila Pujol y Borja Velasco Puntuaciones de propensión y ponderación de probabilidad inversa en la inferencia causal Causal ALGO Bcn

[5] Austin, P. C., Grootendorst, P., & Anderson, G. M.. (2007) *A comparison of the ability of different propensity score models to balance measured variables between treated and untreated subjects: A Monte Carlo study.* Statistics in Medicine, 26, 734–753.

[6] Rosenbaum, P. R. (2002) *Observational studies (2nd ed.)* New York, NY: Springer-Verlag.

[7] Rosenbaum, P. R., Rubin, D.B. (1985) *Constructing a control group using multivariate matched sampling methods that incorporate the propensity score.* The American Statistician, 39, 33–38.

[8] Shamos, Michael (1978). «Computational Geometry». Yale University.

[9] Oliver Kuss, Maria Blettner, Jochen Börgemann (2016) *Propensity Score: an Alternative Method of AnalyzingTreatment Effects.*

[10] Rosenbaum, P. R. (1987a) *Model-based direct ajustment* The journal of the American Statistician, 82, 387-394.

[11] Robins, J. M., Hernan, M. A., Brumback, B. (2000) *Marginal structural models and causal inference in Epi-demiology.* Epidemiology, 11, 550–560.

[12] Stephen L. Morgan, Jennifer J. Todd (2008) *A DIAGNOSTIC ROUTINE FOR THE DETECTION OF CONSEQUENTIAL HETEROGENEITY OF CAUSAL EFFECTS* Sociological Methodology, 231-281.

[13] Cochran, W.G. (1968) *The effectiveness of adjustment by subclassification in removing bias in observational-studies.* Biometrics, 24, 295–313.

[14] Daniel E. Ho, Kosuke Imai, Gary King, Elizabeth A. Stuart (2011). *MatchIt: Nonparametric Preprocessing for Parametric Causal Inference.* Journal of Statistical Software, Vol. 42, No. 8, pp. 1-28.

[15] Kazuki Yoshida and Alexander Bartel (2020). *tableone: Create 'Table 1' to Describe Baseline Characteristics with or without Propensity Score Weights.* R package version 0.12.0.

[16] R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

[17] Tianhui Zhou, Guangyu Tong, Fan Li, Laine E. Thomas and Fan Li (2020) *PSweight: Propensity Score Weighting for Causal Inference with Observational Studies and Randomized Trials.*

[18] James E. Helmreich, Robert M. Pruzek (2009). *PSAgraphics: An R Package to Support Propensity Score Analysis.* Journal of Statistical Software 29(6), 1-23.

[19] Virasakdi Chongsuvivatwong (2018) *epiDisplay: Epidemiological Data Display Package.* R package version 3.5.0.1.

Apéndice A

Código R PS aplicado a Lalonde

Summary de la base de datos

```
lalonde = lalonde[,c(1:3,5:9)]
lalonde$married = as.factor(lalonde$married)
lalonde$nodegree = as.factor(lalonde$nodegree)
lalonde$treat = as.factor(lalonde$treat)
modelo = glm(treat ~ age + educ + married + nodegree
             + re74 + re75, data=lalonde, family=binomial())
summary(modelo)
```

```
##
## Call:
## glm(formula = treat ~ age + educ + married + nodegree + re74 +
##     re75, family = binomial(), data = lalonde)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5998  -0.9185  -0.5286   1.1216   2.9442
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.731e+00  8.267e-01  -3.304 0.000952 ***
## age          2.018e-02  1.093e-02   1.847 0.064806 .
## educ         1.404e-01  5.496e-02   2.554 0.010644 *
## married1    -1.298e+00  2.446e-01  -5.305 1.13e-07 ***
## nodegree1    8.738e-01  2.906e-01   3.007 0.002639 **
## re74        -9.981e-05  2.612e-05  -3.821 0.000133 ***
## re75         6.379e-05  4.127e-05   1.546 0.122118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.49  on 613  degrees of freedom
## Residual deviance: 661.86  on 607  degrees of freedom
## AIC: 675.86
##
## Number of Fisher Scoring iterations: 5
```

```
print(xtable(summary(lalonde)))
```

```
propScore_cov = c("age","educ","married","nodegree","re74", "re75")
tab1 <- CreateTableOne(strata = "treat", vars = propScore_cov,
                       data = lalonde, test = FALSE)
print(tab1, smd = TRUE)
```

```
##                     Stratified by treat
##                      0                1                SMD
##   n                      429              185
##   age (mean (SD))      28.03 (10.79)    25.82 (7.16)     0.242
##   educ (mean (SD))     10.24 (2.86)     10.35 (2.01)     0.045
##   married = 1 (%)        220 (51.3)        35 (18.9)     0.721
##   nodegree = 1 (%)       256 (59.7)       131 (70.8)     0.235
##   re74 (mean (SD)) 5619.24 (6788.75) 2095.57 (4886.62)  0.596
##   re75 (mean (SD)) 2466.48 (3292.00) 1532.06 (3219.25)  0.287
```

Modelo para cada variable

```
modelo1 = lm(re78 ~ treat + age + educ + married + nodegree
             + re74 + re75, data = lalonde)
summary(modelo1)
```

```
modelo2 = glm(treat ~ age + educ + married + nodegree +
                re74 + re75, data = lalonde, family = binomial())
summary(modelo2)
```

```
attach(lalonde)
model1 = lm(re78 ~ age)
summary(model1)
model2 = lm(re78 ~ educ)
summary(model2)
model3 = lm(re78 ~ married)
summary(model3)
model4 = lm(re78 ~ nodegree)
summary(model4)
model5 = lm(re78 ~ re74)
summary(model5)
model6 = lm(re78 ~ re75)
summary(model6)
```

```
model1t = glm(treat ~ age, family=binomial())
summary(model1t)
model2t = glm(treat ~ educ, family=binomial())
summary(model2t)
model3t = glm(treat ~ married, family=binomial())
summary(model3t)
model4t = glm(treat ~ nodegree, family=binomial())
summary(model4t)
model5t = glm(treat ~ re74, family=binomial())
summary(model5t)
model6t = glm(treat ~ re75, family=binomial())
summary(model6t)
```

Calculo de PS

```
lalonde1 = lalonde[lalonde$treat == 1,]
lalonde0 = lalonde[lalonde$treat == 0,]

#Cálculo de PS

form = treat ~ age + married + nodegree + re74 + re75

lalondemodel = glm(form, data = lalonde, family = binomial())

ps = lalondemodel$fitted

newdata = lalonde

newdata$ps = ps
```

```r
ps0 = newdata$ps[newdata$treat == 0]
ps1 = newdata$ps[newdata$treat == 1]

plot(density(newdata$ps), main = "Valores de PS")

plot(density(ps0), main = "Valores de PS para el grupo control")
plot(density(ps1), main = "Valores de PS para el grupo tratamiento")

sumps = c(mean(newdata$ps),sd(newdata$ps))
sumps0 = c(mean(ps0), sd(ps0))
sumps1 = c(mean(ps1), sd(ps1))

sumpss = matrix(c(sumps, sumps0, sumps1), nrow=3, byrow=TRUE,
                dimnames = list(c("PS", "PS_Control",
                                  "PS_Tratamiento"), c("Promedio", "Desviación estándar")))
print(xtable(sumpss))
```

Ilustración de PS matching

```r
model = matchit(treat ~ age + married + nodegree + re74
                + re75, data = lalonde, method = "nearest", ratio = 1)
summary(model)
match = match.data(model)
match$married = as.factor(match$married)
match$nodegree = as.factor(match$nodegree)
match$treat = as.factor(match$treat)
head(match)
dim(match)
plot(model, type="jitter")
propScore_cov = c("age","married","nodegree","re74", "re75")
tab1 <- CreateTableOne(strata = "treat", vars = propScore_cov,
                       data = lalonde, test = FALSE)
print(tab1, smd = TRUE)
tab1m <- CreateTableOne(strata = "treat", vars = propScore_cov,
                        data = match, test = FALSE)
print(tab1m, smd = TRUE)
print(xtable(summary(match)))
```

```r
model = matchit(treat ~ age + married + nodegree + re74
                + re75, data = lalonde, method = "nearest", ratio = 1)
match = match.data(model)
propScore_cov = c("age", "educ", "re74", "re75")

tab1 <- CreateTableOne(strata = "treat", vars = propScore_cov,
                       data = lalonde, test = FALSE)
print(tab1, smd = TRUE)
```

```
##                     Stratified by treat
##                      0                1                SMD
##   n                      429              185
##   age (mean (SD))     28.03 (10.79)    25.82 (7.16)     0.242
##   educ (mean (SD))    10.24 (2.86)     10.35 (2.01)     0.045
```

```
##   re74 (mean (SD)) 5619.24 (6788.75) 2095.57 (4886.62)  0.596
##   re75 (mean (SD)) 2466.48 (3292.00) 1532.06 (3219.25)  0.287
```

```
tab1m <- CreateTableOne(strata = "treat", vars = propScore_cov,
                        data = match, test = FALSE)
print(tab1m, smd = TRUE)
```

```
##                     Stratified by treat
##                      0                 1                 SMD
##   n                      185               185
##   age (mean (SD))     25.69 (11.62)     25.82 (7.16)     0.013
##   educ (mean (SD))     9.81 (2.86)      10.35 (2.01)     0.216
##   re74 (mean (SD)) 2054.21 (3992.22) 2095.57 (4886.62)  0.009
##   re75 (mean (SD)) 1528.87 (2372.42) 1532.06 (3219.25)  0.001
```

```
newmatch = match[order(match$subclass),]
newmatch[c(1:7),]
```

```
##         treat age educ married nodegree     re74     re75        re78  distance
## NSW1        1  37   11       1        1    0.000    0.000  9930.04600 0.2231340
## PSID39      0  49    8       1        1 6459.703 7431.629  7503.89600 0.2242899
## NSW10       1  33   12       1        0    0.000    0.000 12418.07000 0.1613548
## PSID379     0  33   12       1        0    0.000    0.000  5841.45300 0.1613548
## NSW100      1  31    9       0        1    0.000    0.000 26817.60000 0.5020001
## PSID196     0  18   11       0        1    0.000 1367.806    33.98771 0.4769648
## NSW101      1  24   10       0        1    0.000    0.000     0.00000 0.4770932
##         weights subclass
## NSW1          1        1
## PSID39        1        1
## NSW10         1        2
## PSID379       1        2
## NSW100        1        3
## PSID196       1        3
## NSW101        1        4
```

```
matchnew = merge(match, lalonde)
modelo1 = glm(re78 ~ treat + age + educ + married + nodegree + re74
              + re75, data = matchnew[c(1:370),], family = gaussian(link = identity))
summary(modelo1)
```

```
##
## Call:
## glm(formula = re78 ~ treat + age + educ + married + nodegree +
##     re74 + re75, family = gaussian(link = identity), data = matchnew[c(1:370),
##     ])
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -12051   -4607   -1777    2679   54728
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1631.7919  3077.2911   0.530   0.5963
## treat1       201.0151   726.5990   0.277   0.7822
## age          -49.6586    42.7233  -1.162   0.2459
## educ         478.7875   204.7547   2.338   0.0199 *
```

```
## married1     701.1460  1007.4229   0.696   0.4869
## nodegree1   -596.3756  1111.7212  -0.536   0.5920
## re74           0.1039     0.1020   1.019   0.3091
## re75           0.3637     0.1593   2.283   0.0230 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 47058569)
##
##     Null deviance: 1.8614e+10  on 369  degrees of freedom
## Residual deviance: 1.7035e+10  on 362  degrees of freedom
## AIC: 7596.7
##
## Number of Fisher Scoring iterations: 2
```

Ilustración de PS por ponderación

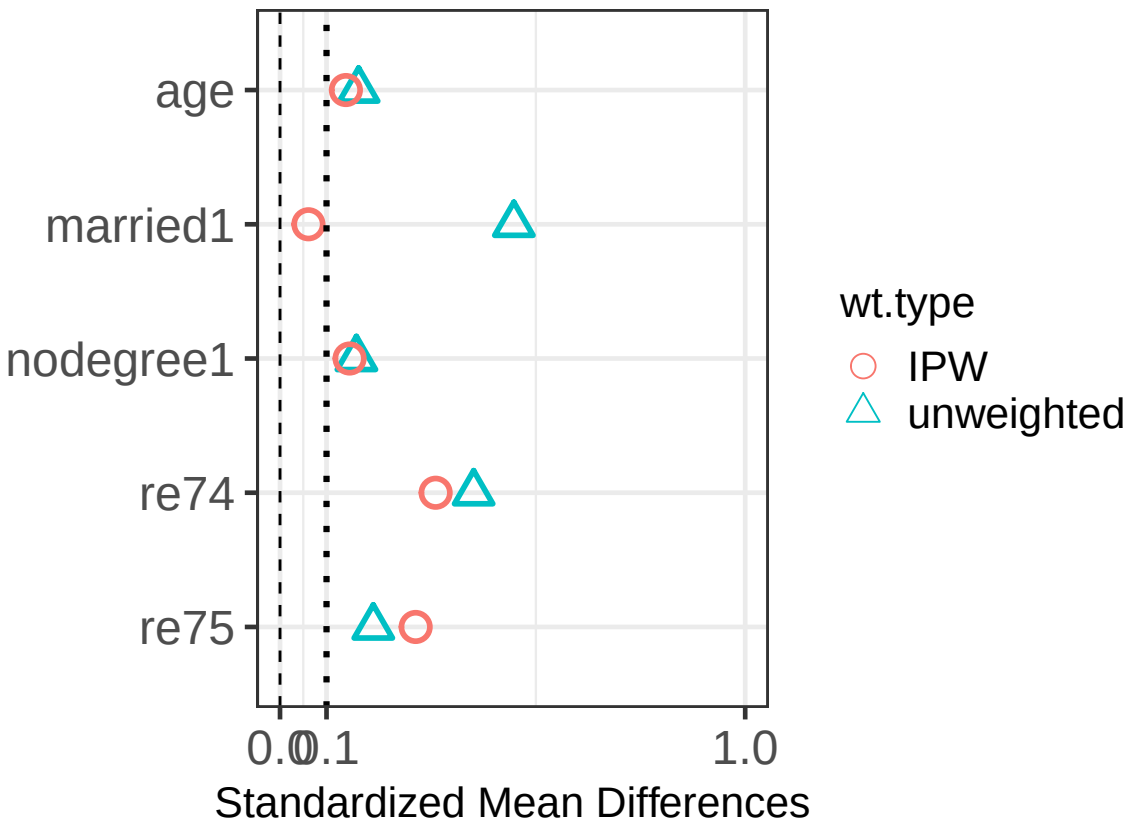
```
library(PSweight)
# Generar el modelo
ps.any = treat ~ age + married + nodegree + re74 + re75
# Generar gráficas de distribución del PS generado por ponderación
bal.any = SumStat(ps.formula = ps.any, data=lalonde,
                  weight = c("IPW"))
bal.any
```

```
## weights estimated for:  IPW
```

```
summary(bal.any)
```

```
## unweighted result
##             Mean 0   Mean 1   SMD
## age         28.030   25.816 0.242
## married1     0.513    0.189 0.719
## nodegree1    0.597    0.708 0.235
## re74      5619.237 2095.574 0.596
## re75      2466.484 1532.055 0.287
##
## IPW result
##             Mean 0   Mean 1   SMD
## age         27.301   26.089 0.135
## married1     0.415    0.446 0.069
## nodegree1    0.630    0.557 0.155
## re74      4588.482 7782.340 0.540
## re75      2198.306 3371.090 0.360
```

```
plot(bal.any, type="balance", metric = "PSD")
```




```
# Estimamos el promedio de la variable resultado con los
# diversos grupos generados con el PS, que los genera automaticamente esta función
ate.any = PSweight(ps.formula = ps.any, zname = "treat",
                   yname = "re78", data = lalonde, weight = "IPW")

ate.any
```

```
## Original group value:  0, 1
##
## Point estimate:
## 6534.3597, 10083.8602
```

```
# Miramos si las medias de ambos grupos son iguales o diferentes
summary(ate.any, CI = TRUE)
```

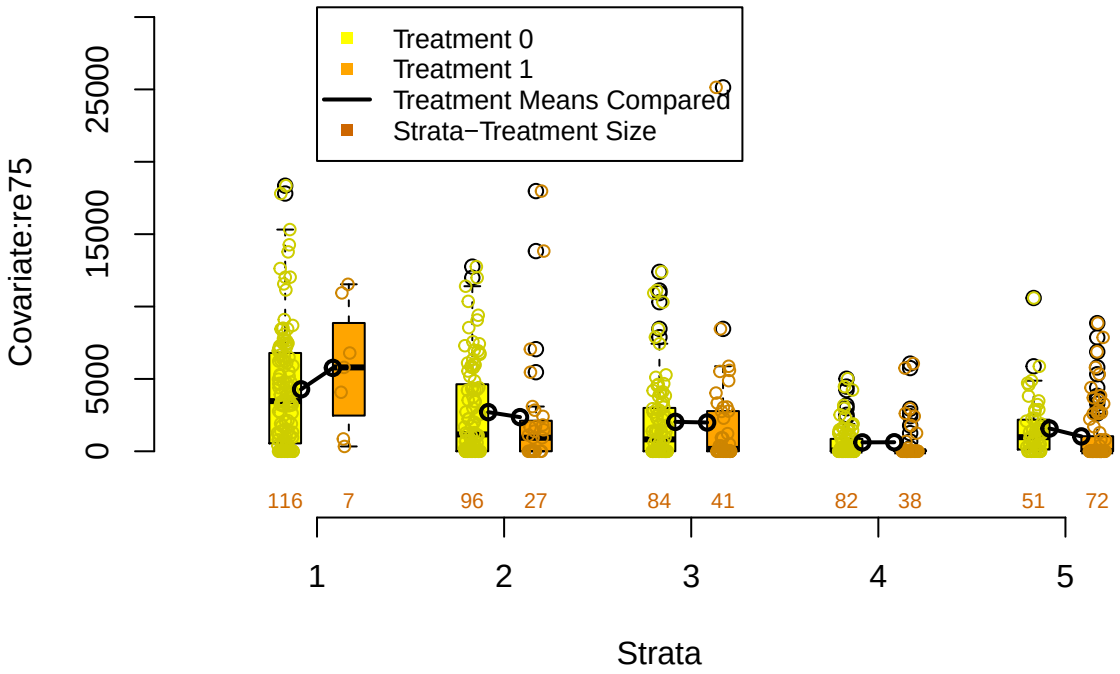
```
##
## Closed-form inference:
##
## Original group value:  0, 1
##
## Contrast:
##              0 1
## Contrast 1 -1 1
##
##            Estimate Std.Error     lwr    upr Pr(>|z|)
## Contrast 1   3549.5    3189.8 -2702.4 9801.5   0.2658
```

```
newdata = lalonde
newdata$ps = bal.any$propensity[,1]
```

Ilustración con PS por estratificación

```
modelo = glm(treat ~ age + married + nodegree + re74
             + re75, data = lalonde, family = binomial())

attach(lalonde)
#Generamos los valores de PS
ps = modelo$fitted
#Estratificamos por 5 valores y 10 valores
modelo.s5 <- cut(ps, quantile(ps, seq(0, 1, 1/5)),
                 include.lowest = TRUE, labels = FALSE)


# Ver como quedan los grupos cuando se
# separan por estratificación en covariables continuas
box.psa(re75, treat, modelo.s5, xlab = "Strata",
        ylab = "Covariate:re75", balance = FALSE)
```

```
## Warning in xy.coords(x, y): NAs introducidos por coerción

## Warning in xy.coords(x, y): NAs introducidos por coerción
```



```r
box.psa(re74, treat, modelo.s5, xlab = "Strata",
        ylab = "Covariate:re74", balance = FALSE)
box.psa(age, treat, modelo.s5, xlab = "Strata",
        ylab = "Covariate:age", balance = FALSE)


# Ver como quedan los grupos cuando se separan por estratificación en covariables categóricas
cat.psa(married, treat, modelo.s5, xlab = "Strata",
        ylab = "Proportion for 'married'", catnames
        = c("No married", "Married"), barnames = c("No Treat",
                                                   "With Treat"), rtmar = 2)
cat.psa(nodegree, treat, modelo.s5, xlab = "Strata",
        ylab = "Proportion for 'nodegree'", catnames = c("No Degree", "Degree")
        , barnames = c("No Treat", "With Treat"), rtmar = 2)




# En la base de datos, añadir una columna con
# el número correspondiente al estrato al que pertenecen
newdata = lalonde
newdata$strata = modelo.s5
```

Ilustración del PS por regresión de covariables

```r
# Primero generamos el modelo para el tratamiento

modelo = glm(treat ~ age + married + nodegree
             + re74 + re75, data = lalonde, family = binomial())

# Generamos los valores de PS

ps = modelo$fitted

# Añadirlo a la base de datos

lalonde$ps = ps

newdata = lalonde

modelo1 = lm(re78 ~ treat + ps, data = newdata)
modelo2 = lm(re78 ~ treat, data=newdata)
modelo3 = lm(re78 ~ treat + age + married + nodegree + re74 + re75, data=newdata)

# Realizar el estudio para ver si ps es significativo con la función logistic.display
#install.packages("epiDisplay")

#logistic.display(modelo)

summary(modelo1)
```

```
##
## Call:
```

```
## lm(formula = re78 ~ treat + ps, data = newdata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9903  -5351  -1863   3859  55339
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9935.5      624.8  15.901  < 2e-16 ***
## treat1         748.3      685.0   1.093    0.275
## ps          -11178.7     1956.0  -5.715 1.72e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7285 on 611 degrees of freedom
## Multiple R-squared:  0.05219,    Adjusted R-squared:  0.04909
## F-statistic: 16.82 on 2 and 611 DF,  p-value: 7.732e-08
```

```
summary(modelo2)
```

```
##
## Call:
## lm(formula = re78 ~ treat, data = newdata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6984  -6349  -2048   4100  53959
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6984.2      360.7  19.362   <2e-16 ***
## treat1        -635.0      657.1  -0.966    0.334
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7471 on 612 degrees of freedom
## Multiple R-squared:  0.001524,   Adjusted R-squared:  -0.0001079
## F-statistic: 0.9338 on 1 and 612 DF,  p-value: 0.3342
```

```
summary(modelo3)
```

```
##
## Call:
## lm(formula = re78 ~ treat + age + married + nodegree + re74 +
##     re75, data = newdata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14031  -4858  -1666   3643  55163
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.409e+03  9.690e+02   5.582 3.58e-08 ***
## treat1       9.693e+02  6.541e+02   1.482   0.1389
## age         -8.024e+00  3.172e+01  -0.253   0.8004
```

```
## married1     4.337e+02  6.889e+02   0.629   0.5293
## nodegree1   -1.292e+03  6.008e+02  -2.150   0.0319 *
## re74         3.213e-01  5.793e-02   5.546 4.36e-08 ***
## re75         2.201e-01  1.048e-01   2.101   0.0361 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6987 on 607 degrees of freedom
## Multiple R-squared:  0.1338, Adjusted R-squared:  0.1252
## F-statistic: 15.63 on 6 and 607 DF,  p-value: < 2.2e-16
```

Apéndice B

Código R PS aplicado a NHANES

Cargar los datos NHANES

Basado en:

```
#https://www.kaggle.com/wildscop/accessing-nhanes-data-directly-from-cdc-website
```

```
list.of.packages <- c("nhanesA", "DataExplorer")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
library(nhanesA)
```

Variables de los datos DEMO_J

```
# Demographics component
dd <- nhanes('DEMO_J') # Demographics Data file name for NHANES 2017-2018
# nhanes() function downloads NHANES data in SAS format.
# File name obtained searching wwwn.cdc.gov/nchs/nhanes/search/default.aspx
# Identify variable names by checking Codebook and Data Documentation
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO_J.htm
dd.vars <- c("SEQN", "RIAGENDR", "RIDAGEYR", "RIDRETH1", "DMDBORN4", "DMDCITZN","DMDYRSUS",
             "DMDEDUC3", "DMDEDUC2", "DMDMARTL", "INDHHIN2", "INDFMPIR", "RIDEXPRG",
             "WTINT2YR", "WTMEC2YR", "SDMVPSU", "SDMVSTRA")
# Feel free to add a few more variables that seems relevant
dd.data <- nhanesTranslate('DEMO_J', dd.vars, data=dd[,dd.vars])
# nhanesTranslate() function recoded categorical variable labels
names(dd.data) <- c("id", "gender", "age", "race", "born", "citizen", "lenghthinUS", "eduyouth",
                    "education", "marriage", "annualincome", "incometopoverty", "pregnancy",
                     "interview.weight", "MEC.weight", "PSU", "strata")
# rename variable names so that they can be identified easily later
```

```
dim(dd.data)
table(dd.data$gender, useNA = "always")
table(dd.data$race, useNA = "always")
table(dd.data$education, useNA = "always")
# Compare with the frequency table in https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO_J.htm
```

Variables de los datos MCQ_J

```
# Medical Conditions
md <- nhanes('MCQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/MCQ_J.htm
md.vars <- c("SEQN", "MCQ053","MCQ160A","MCD180A","MCQ195",
             "MCQ160E","MCD180E","MCQ300A")
md.data <- nhanesTranslate('MCQ_J', md.vars, data=md[,md.vars])
names(md.data) <- c("id", "treat.anemia", "arthritis.has", "arthritis.diag.age",
                    "arthritis.type", "heart.attack",
                    "heart.attack.diag.age", "heart.attack.relative")
summary(md.data$arthritis.diag.age)
# note that there is an age record of '99999'
# (check codebook) '99999' is the code for 'Don't know'
```

Variables de los demás datos

```
# Body Measures
bd <- nhanes('BMX_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/BMX_J.htm
bd.vars <- c("SEQN", "BMXBMI")
bd.data <- nhanesTranslate('BMX_J', bd.vars, data=bd[,bd.vars])
names(bd.data) <- c("id", "bmi")

# Diabetes
dqd <- nhanes('DIQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DIQ_J.htm
dqd.vars <- c("SEQN", "DIQ010", "DID040")
dqd.data <- nhanesTranslate('DIQ_J', dqd.vars, data=dqd[,dqd.vars])
names(dqd.data) <- c("id", "diabetes", "diabetes.diag.age")

# Smoking
sd <- nhanes('SMQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/SMQ_J.htm
sd.vars <- c("SEQN", "SMQ020", "SMQ040", "SMQ050Q")
sd.data <- nhanesTranslate('SMQ_J', sd.vars, data=sd[,sd.vars])
names(sd.data) <- c("id", "smoke.life",  "smoke.now", "smoke.quit")

# Alcohol Use
ad <- nhanes('ALQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/ALQ_J.htm
ad.vars <- c("SEQN", "ALQ121", "ALQ111", "ALQ130")
ad.data <- nhanesTranslate('ALQ_J', ad.vars, data=ad[,ad.vars])
names(ad.data) <- c("id", "Alcohol.1.yr",  "Alcohol.life",
                    "Alcohol.freq1.yr")

# Physical Activity
pd <- nhanes('PAQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/PAQ_J.htm
pd.vars <- c("SEQN", "PAQ605", "PAQ620")
pd.data <- nhanesTranslate('PAQ_J', pd.vars, data=pd[,pd.vars])
names(pd.data) <- c("id", "physical.vigorous",  "physical.moderate")

# Blood Pressure and Cholesterol
htd <- nhanes('BPQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/BPQ_J.htm
htd.vars <- c("SEQN", "BPQ020", "BPQ030", "BPD035", "BPQ050A",
              "BPQ040A", "BPQ080")
htd.data <- nhanesTranslate('BPQ_J', htd.vars, data=htd[,htd.vars])
names(htd.data) <- c("id", "blood.pressure.ever", "blood.pressure.2",
                     "age.hypertension",
                     "blood.pressure.med", "Prehypertension",
                     "high cholesterol.high")
```

```r
# Hospital Utilization and Access to Care
hd <- nhanes('HUQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/HUQ_E.htm
hd.vars <- c("SEQN", "HUQ030")
hd.data <- nhanesTranslate('HUQ_J', hd.vars, data=hd[,hd.vars])
names(hd.data) <- c("id", "medical.access")

# Health Insurance
id <- nhanes('HIQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/HIQ_J.htm
id.vars <- c("SEQN", "HIQ011")
id.data <- nhanesTranslate('HIQ_J', id.vars, data=id[,id.vars])
names(id.data) <- c("id", "covered.health")

# Diet Behavior and Nutrition
hdd <- nhanes('DBQ_J')
# Check codebook at
# https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DBQ_J.htm
hdd.vars <- c("SEQN", "DBQ700")
hdd.data <- nhanesTranslate('DBQ_J', hdd.vars, data=hdd[,hdd.vars])
names(hdd.data) <- c("id", "healthy.diet")
```

Unir la base de datos

```r
# merge by id
merged.data <- merge(md.data, dd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, bd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, dqd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, ad.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, sd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, pd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, hd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, htd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, hdd.data, by = c("id"), all=TRUE)
merged.data <- merge(merged.data, id.data, by = c("id"), all=TRUE)
dim(merged.data)
save(merged.data, file="NHANES2018.RData")
```

```r
int.count<-sapply(merged.data,function(x)  is.integer(x))
int.count[int.count==TRUE]
```

```r
summary(merged.data$treat.anemia)
```

Limpieza de datos

```r
summary(merged.data$arthritis.diag.age)
```

```r
merged.data$arthritis.diag.age[merged.data$arthritis.diag.age == 99999] <- NA
# 99999 is code for "Don't know"
```

```r
summary(merged.data$arthritis.diag.age)
```

```r
summary(merged.data$heart.attack.diag.age)
```

```r
merged.data$heart.attack.diag.age[merged.data$heart.attack.diag.age == 99999] <- NA
# 99999 is code for "Don't know"
summary(merged.data$heart.attack.diag.age)
```

```r
summary(merged.data$diabetes.diag.age)
```

```r
merged.data$diabetes.diag.age[merged.data$diabetes.diag.age == 999] <- NA
# 999 is code for   "Don't know"
merged.data$diabetes.diag.age[merged.data$diabetes.diag.age == 666] <- 1
# 666   is code for 'Less than 1 year'
summary(merged.data$diabetes.diag.age)
```

```r
summary(merged.data$age.hypertension)
```

```r
merged.data$age.hypertension[merged.data$age.hypertension == 999] <- NA
# 999 is code for   "Don't know"
summary(merged.data$age.hypertension)
```

```r
invalid.exclude <- function(x, exclude=c("Other", "Refused", "Don't know")) {
  x <- as.character(x)
  x[x %in% exclude] <- NA
  return(x)
}
require(data.table)
DT2 <- data.table(merged.data)

table(DT2$arthritis.has)
```

```r
DT2$treat.anemia <- invalid.exclude(DT2$treat.anemia)
table(DT2$treat.anemia)
```

```r
DT2$arthritis.has <- invalid.exclude(DT2$arthritis.has)
table(DT2$arthritis.has)
```

```r
DT2$arthritis.diag.age[DT2$arthritis.has == "No"] <- 987
DT2$heart.attack.diag.age[DT2$heart.attack == "No"] <- 987
DT2$arthritis.type <- as.character(DT2$arthritis.type)
DT2$arthritis.type[DT2$arthritis.has == "No"] <- "Non-arthritis"

table(DT2$arthritis.type)
```

```r
DT2$arthritis.type <- invalid.exclude(DT2$arthritis.type)
table(DT2$arthritis.type)
```

```r
table(DT2$heart.attack)
DT2$heart.attack <- invalid.exclude(DT2$heart.attack)
table(DT2$heart.attack)
```

```r
DT3 <- na.omit(DT2, cols=c("arthritis.type","heart.attack"))
dim(DT3)
```

```r
DT4 <- DT3[(DT3$arthritis.diag.age <= DT3$heart.attack.diag.age) |
             DT3$arthritis.type=="Non-arthritis",]
```

```
dim(DT4)
```

```
summary(DT4$diabetes)
```

```
DT4$diabetes <- invalid.exclude(DT4$diabetes, exclude=c("Borderline", "Don't know"))
DT4$diabetes.diag.age[DT4$diabetes == "No"] <- 987
table(DT4$diabetes)
```

```
DT5 <- DT4[(DT4$diabetes.diag.age <= DT4$heart.attack.diag.age) |
             (DT4$diabetes=="No" | is.na(DT4$diabetes)),]
dim(DT5)
```

```
DT5 <- DT5[(DT5$diabetes.diag.age <= DT5$arthritis.diag.age) |
             (DT5$diabetes=="No" | is.na(DT5$diabetes)),]
dim(DT5)
```

```
with(DT5, table(arthritis.type, heart.attack))
```

```
summary(DT5$gender)
```

```
summary(DT5$age)
```

```
DT5$age <- cut(DT5$age, breaks = c(19.9, 50, 70, 81))
table(DT5$age)
```

```
DT5$bmi <- cut(DT5$bmi, breaks = c(0, 25, 80))
levels(DT5$age)[levels(DT5$age)=="(70,80]"] <- "70+"
summary(DT5$age)
```

```
summary(DT5$race)
```

```
require(car)
summary(DT5$race)
```

```
DT5$race <- car:::recode(DT5$race,
                         "'Non-Hispanic White'='White';
                         'Non-Hispanic Black'='Black';
                         c('Other Race - Including Multi-Rac', 'Mexican American','Other Hispanic')='Othe
                         else=NA")
summary(DT5$race)
```

```
table(DT5$born)
```

```
DT5$born <- car:::recode(DT5$born,
                         "'Born in 50 US states or Washingt'='USborn';
                         'Others'='Other';
                         else=NA")
table(DT5$born)
```

```
table(DT5$education)
```

```
DT5$education <- car:::recode(DT5$education,
                              "c('Some college or AA degree',
                                 'College graduate or above')='College';
                               c('9-11th grade (Includes 12th grad',
                                 'High school graduate/GED or equi')='High.School';
                               'Less than 9th grade'='School';
                               else=NA")
```

```
table(DT5$education)
```

```
table(DT5$marriage)
```

```
DT5$marriage <- car:::recode(DT5$marriage,
                       "c('Never married','Divorced','Separated','Widowed')='Not.married';
                        c('Living with partner', 'Married')='Married';
                        else=NA")
table(DT5$marriage)
```

```
table(DT5$annualincome)
```

```
DT5$annualincome <- car:::recode(DT5$annualincome,
                                 "c('$     0 to $ 4,999', '$ 5,000 to $ 9,999',
                                    '$10,000 to $14,999', '$15,000 to $19,999')='<20k';
                                 c('$20,000 to $24,999', '$25,000 to $34,999',
                                    '$35,000 to $44,999',
                                   '$45,000 to $54,999') = '20kto54k';
                                 c('$55,000 to $64,999', '$65,000 to $74,999',
                                   '$75,000 to $99,999','$100,000 and Over')
                                  = '55k+';
                                 else=NA")
table(DT5$annualincome)
```

```
table(DT5$smoke.life)
```

```
DT5$smoke <- invalid.exclude(DT5$smoke.life)
table(DT5$smoke)
```

```
DT5$physical.vigorous <- invalid.exclude(DT5$physical.vigorous)
table(DT5$physical.vigorous)
```

```
DT5$physical.moderate <- invalid.exclude(DT5$physical.moderate)
table(DT5$physical.moderate)
```

```
physical.activity <- rep("No", length(DT5$physical.vigorous))
physical.activity[DT5$physical.moderate=="Yes"] <- "Moderate"
physical.activity[DT5$physical.vigorous=="Yes"] <- "High"
DT5$physical.activity <- invalid.exclude(physical.activity)
table(DT5$physical.activity)
```

```
table(DT5$medical.access)
```

```
DT5$medical.access <- car:::recode(DT5$medical.access,
                       "c('Yes','There is more than one place')='Yes';
                        'There is no place'='No';
                        else=NA")
table(DT5$medical.access)
```

```
table(DT5$blood.pressure.ever)
```

```
DT5$blood.pressure <- invalid.exclude(DT5$blood.pressure.ever)
table(DT5$blood.pressure)
```

```
table(DT5$healthy.diet)
```

```
DT5$healthy.diet <- car:::recode(DT5$healthy.diet,
                        "c('Excellent','Very good','Good')='Good';
                         'Fair'='Fair';
                         'Poor'='Poor';
                         else=NA")
table(DT5$healthy.diet)
```

```
table(DT5$covered.health)
```

```
DT5$covered.health <- invalid.exclude(DT5$covered.health)
table(DT5$covered.health)
```

Resumen de la base de datos NHANES

```
DT6 <- DT5[,c("arthritis.type", "treat.anemia", "heart.attack", "gender",
              "interview.weight", "MEC.weight", "PSU", "strata",
              "bmi", "diabetes", "smoke", "age",
              "race", "born", "education", "marriage",
              "annualincome", "physical.activity", "medical.access",
              "blood.pressure", "healthy.diet", "covered.health")]
# No. of missing values in each variable
sort(sapply(DT6,function(x) sum(is.na (x))), decreasing = TRUE)
```

```
DT7 <- subset(DT6, arthritis.type != "Osteoarthritis")
dim(DT7)
```

```
require(DataExplorer)
plot_missing(DT7)
```

```
DT8 <- na.omit(DT7)
dim(DT8)
```

```
# (for simplicity, and to allow more data).
# require(DataExplorer)
# plot_missing(DT6)
# require(mice)
# imp <- mice(DT6, m = 1, maxit = 1,seed = 1, print = FALSE)
# DT7 <- complete(imp)
# plot_missing(DT7)
# DT8 <- na.omit(DT7)
# dim(DT8)
```

```
DT8$treat.anemia <- as.factor(DT8$treat.anemia)
DT8$heart.attack <- as.factor(DT8$heart.attack)
DT8$heart.attack <- with(DT8, relevel(heart.attack, ref = "No"))
DT8$race <- with(DT8, relevel(race, ref = "White"))
DT8$healthy.diet <- with(DT8, relevel(healthy.diet, ref = "Poor"))
DT8$education <- with(DT8, relevel(education, ref = "School"))
DT8$marriage <- with(DT8, relevel(marriage, ref = "Not.married"))
DT8$physical.activity <- as.factor(DT8$physical.activity)
DT8$physical.activity <- with(DT8, relevel(physical.activity, ref = "No"))

require(tableone)
tab1 <- CreateTableOne(data = DT8, includeNA = TRUE,
```

```
                        strata = "heart.attack", test = FALSE,
                        var = c("arthritis.type", "gender",
                                "bmi", "diabetes", "smoke", "age",
                                "race", "born", "education", "marriage",
                                "annualincome", "physical.activity", "medical.access",
                                "blood.pressure", "healthy.diet", "covered.health"))
print(tab1,showAllLevels = FALSE)
```

```
table(DT8$arthritis.type)
```

```
table(DT8$heart.attack)
```

```
with(DT8, table(arthritis.type, heart.attack))
```

```
### Exposure is Rheumatoid arthritis
DT9a <- subset(DT8,arthritis.type != "Osteoarthritis")
tab2 <- CreateTableOne(data = DT9a, includeNA = TRUE,
                        strata = "arthritis.type", test = FALSE,
                        var = c("heart.attack", "gender",
                                "bmi", "diabetes", "smoke", "age",
                                "race", "born", "education", "marriage",
                                "annualincome", "physical.activity", "medical.access",
                                "blood.pressure", "healthy.diet", "covered.health"))
print(tab2,showAllLevels = FALSE, smd = TRUE)
```

```
cross.tab <- with(DT9a, table(arthritis.type, heart.attack))
cross.tab
```

```
save(DT6, DT8, DT9a, file="analyticNHANES2018.RData")
dim(DT9a)
```

Análisis de la base de datos NHANES

```
summary(DT9a[,c("heart.attack", "arthritis.type", "gender",
                "bmi", "diabetes", "smoke", "age", "marriage",
                "annualincome", "physical.activity",
                "medical.access", "blood.pressure", "healthy.diet")])
```

```
##  heart.attack       arthritis.type    gender          bmi       diabetes
##  No :3439     Non-arthritis:2883   Male  :1734   (0,25] :1004   No :3124
##  Yes: 110     Yes-arthritis: 666   Female:1815   (25,80]:2545   Yes: 425
##
##  smoke             age              marriage      annualincome
##  No :2114   (19.9,50]:1857   Not.married:1354   <20k    : 521
##  Yes:1435   (50,70]  :1197   Married    :2195   20kto54k:1385
##             (70,81]  : 495                      55k+    :1643
##  physical.activity medical.access blood.pressure healthy.diet
##  No      :1841     No : 701       No :2362       Poor: 229
##  High    : 897     Yes:2848       Yes:1187       Fair: 920
##  Moderate: 811                                   Good:2400
```

```
print(xtable(summary(DT9a[,c("heart.attack", "arthritis.type",
                             "gender", "bmi", "diabetes", "smoke", "age")])))
print(xtable(summary(DT9a[,c("marriage", "annualincome",
                             "physical.activity", "medical.access",
                             "blood.pressure", "healthy.diet")])))
```

```
cov1 = c("arthritis.type", "gender", "bmi", "diabetes", "smoke",
         "age", "marriage", "annualincome", "physical.activity",
         "medical.access", "blood.pressure", "healthy.diet")
tab1 = compareGroups(heart.attack ~ arthritis.type + gender + bmi
                     + diabetes + smoke + age + marriage + annualincome
                     + physical.activity + medical.access + blood.pressure
                     + healthy.diet, data=DT9a)
createTable(tab1)
```

```
##
## --------Summary descriptives table by 'heart.attack'---------
##
## _____________________________________________________
##                         No          Yes     p.overall
##                       N=3439       N=110
## _____________________________________________________
## arthritis.type:                                 0.001
##     Non-arthritis  2807 (81.6%) 76 (69.1%)
##     Yes-arthritis  632 (18.4%)  34 (30.9%)
## gender:                                        <0.001
##     Male           1651 (48.0%) 83 (75.5%)
##     Female         1788 (52.0%) 27 (24.5%)
## bmi:                                            0.573
##     (0,25]         976 (28.4%)  28 (25.5%)
##     (25,80]        2463 (71.6%) 82 (74.5%)
## diabetes:                                      <0.001
##     No             3048 (88.6%) 76 (69.1%)
##     Yes            391 (11.4%)  34 (30.9%)
## smoke:                                         <0.001
```

```
##     No              2069 (60.2%) 45 (40.9%)
##     Yes             1370 (39.8%) 65 (59.1%)
## age:                                        <0.001
##     (19.9,50]       1848 (53.7%)  9 (8.18%)
##     (50,70]         1145 (33.3%) 52 (47.3%)
##     (70,81]          446 (13.0%)  49 (44.5%)
## marriage:                                    0.270
##     Not.married     1306 (38.0%) 48 (43.6%)
##     Married         2133 (62.0%) 62 (56.4%)
## annualincome:                               <0.001
##     <20k             485 (14.1%)  36 (32.7%)
##     20kto54k        1338 (38.9%) 47 (42.7%)
##     55k+            1616 (47.0%) 27 (24.5%)
## physical.activity:                           0.394
##     No              1777 (51.7%) 64 (58.2%)
##     High             872 (25.4%)  25 (22.7%)
##     Moderate         790 (23.0%)  21 (19.1%)
## medical.access:                              0.006
##     No               691 (20.1%)  10 (9.09%)
##     Yes             2748 (79.9%) 100 (90.9%)
## blood.pressure:                             <0.001
##     No              2329 (67.7%) 33 (30.0%)
##     Yes             1110 (32.3%) 77 (70.0%)
## healthy.diet:                                0.001
##     Poor             214 (6.22%)  15 (13.6%)
##     Fair             903 (26.3%)  17 (15.5%)
##     Good            2322 (67.5%) 78 (70.9%)
## ------------------------------------------------------
```

```r
cov2 = c("gender", "bmi", "diabetes", "smoke", "age", "marriage",
         "annualincome", "physical.activity", "medical.access",
         "blood.pressure", "healthy.diet")
tab2 = CreateCatTable(strata = "arthritis.type", vars = cov2, data=DT9a, test=FALSE)
print(tab2, smd=TRUE)
```

```
##                           Stratified by arthritis.type
##                            Non-arthritis Yes-arthritis SMD
##   n                        2883          666
##   gender = Female (%)      1403 (48.7)   412 (61.9)     0.268
##   bmi = (25,80] (%)        2018 (70.0)   527 (79.1)     0.211
##   diabetes = Yes (%)        315 (10.9)   110 (16.5)     0.163
##   smoke = Yes (%)          1098 (38.1)   337 (50.6)     0.254
##   age (%)                                               0.978
##      (19.9,50]             1734 (60.1)   123 (18.5)
##      (50,70]                866 (30.0)   331 (49.7)
##      (70,81]                283 ( 9.8)   212 (31.8)
##   marriage = Married (%)   1803 (62.5)   392 (58.9)     0.075
##   annualincome (%)                                      0.226
##      <20k                   383 (13.3)   138 (20.7)
##      20kto54k              1118 (38.8)   267 (40.1)
##      55k+                  1382 (47.9)   261 (39.2)
##   physical.activity (%)                                0.134
##      No                    1493 (51.8)   348 (52.3)
##      High                   755 (26.2)   142 (21.3)
##      Moderate               635 (22.0)   176 (26.4)
```

```
##   medical.access = Yes (%) 2233 (77.5)   615 (92.3)     0.425
##   blood.pressure = Yes (%)  794 (27.5)   393 (59.0)     0.670
##   healthy.diet (%)                                      0.054
##      Poor                   182 ( 6.3)    47 ( 7.1)
##      Fair                   738 (25.6)   182 (27.3)
##      Good                  1963 (68.1)   437 (65.6)
```

```
model = glm(heart.attack ~ arthritis.type+gender+bmi+
              diabetes+smoke+age+marriage+
              annualincome+physical.activity+medical.access+
              blood.pressure+healthy.diet, family = binomial())
summary(model)
```

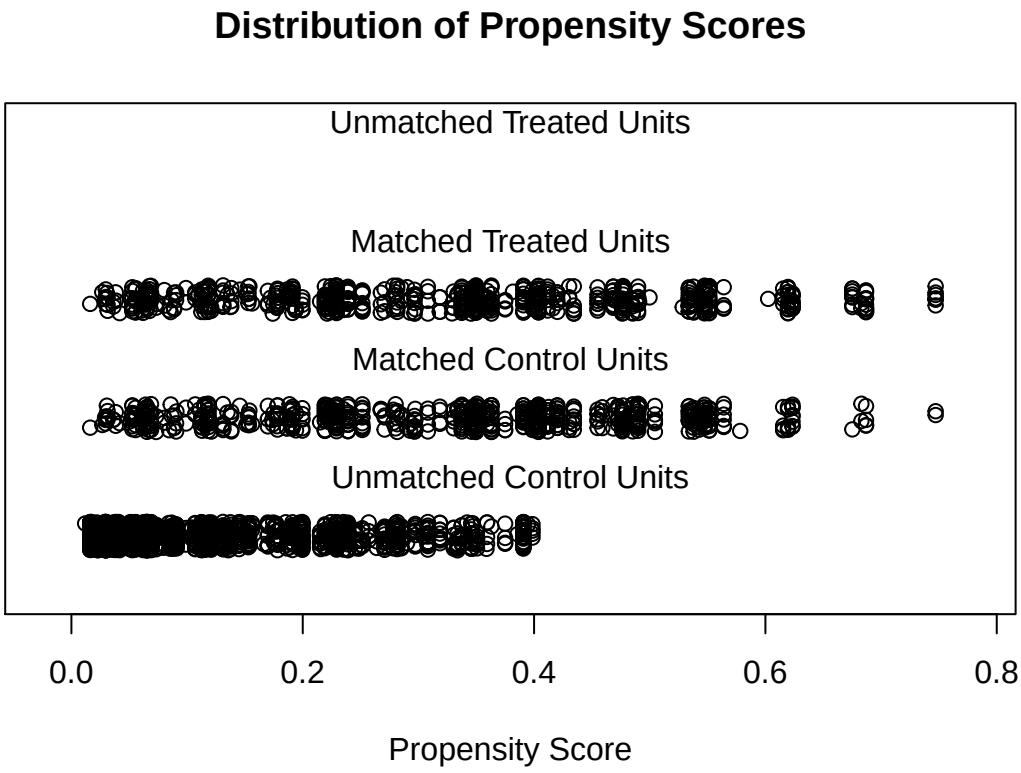
```
##
## Call:
## glm(formula = heart.attack ~ arthritis.type + gender + bmi +
##     diabetes + smoke + age + marriage + annualincome + physical.activity +
##     medical.access + blood.pressure + healthy.diet, family = binomial())
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1930  -0.2468  -0.1241  -0.0701   3.3854
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -3.792956   0.564533  -6.719 1.83e-11 ***
## arthritis.typeYes-arthritis -0.027983   0.232426  -0.120 0.904171
## genderFemale                 -1.192023   0.251968  -4.731 2.24e-06 ***
## bmi(25,80]                    0.008556   0.244280   0.035 0.972061
## diabetesYes                   0.370740   0.236839   1.565 0.117497
## smokeYes                      0.119466   0.218784   0.546 0.585035
## age(50,70]                    1.776851   0.381778   4.654 3.25e-06 ***
## age(70,81]                    2.423257   0.403913   5.999 1.98e-09 ***
## marriageMarried              -0.074760   0.221038  -0.338 0.735194
## annualincome20kto54k         -0.684074   0.247754  -2.761 0.005761 **
## annualincome55k+             -1.208005   0.290639  -4.156 3.23e-05 ***
## physical.activityHigh         0.052041   0.259681   0.200 0.841167
## physical.activityModerate    -0.178229   0.269397  -0.662 0.508238
## medical.accessYes             0.398918   0.359517   1.110 0.267174
## blood.pressureYes             0.835981   0.238714   3.502 0.000462 ***
## healthy.dietFair             -1.303381   0.384843  -3.387 0.000707 ***
## healthy.dietGood             -0.799307   0.320249  -2.496 0.012564 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 980.82  on 3548  degrees of freedom
## Residual deviance: 775.55  on 3532  degrees of freedom
## AIC: 809.55
##
## Number of Fisher Scoring iterations: 8
```

Ilustración de PS matching

```
PSformula2 = arthritis.type ~ gender + diabetes + smoke + age +
  annualincome + medical.access + blood.pressure
model = matchit(formula = PSformula2, data = DT9a, method = "nearest", ratio = 1)
#summary(model)
match = match.data(model)
dim(match)
```

```
## [1] 1332   25
```

```
plot(model, type="jitter")
```




```
## [1] "To identify the units, use first mouse button; to stop, use second."
```

```
## integer(0)
```

```
require(tableone)
baselinevars <- c("gender","diabetes", "smoke", "age",
                  "annualincome", "medical.access",
                  "blood.pressure")
tab1 <- CreateTableOne(strata = "arthritis.type", vars = baselinevars,
                       data = DT9a, test = FALSE)
print(tab1, smd = TRUE)
```

```
##                          Stratified by arthritis.type
##                           Non-arthritis Yes-arthritis SMD
##   n                        2883          666
```

```
##   gender = Female (%)      1403 (48.7)   412 (61.9)     0.268
##   diabetes = Yes (%)        315 (10.9)   110 (16.5)     0.163
##   smoke = Yes (%)          1098 (38.1)   337 (50.6)     0.254
##   age (%)                                               0.978
##      (19.9,50]             1734 (60.1)   123 (18.5)
##      (50,70]                866 (30.0)   331 (49.7)
##      (70,81]                283 ( 9.8)   212 (31.8)
##   annualincome (%)                                      0.226
##      <20k                    383 (13.3)   138 (20.7)
##      20kto54k               1118 (38.8)   267 (40.1)
##      55k+                   1382 (47.9)   261 (39.2)
##   medical.access = Yes (%) 2233 (77.5)   615 (92.3)     0.425
##   blood.pressure = Yes (%)  794 (27.5)   393 (59.0)     0.670
```

```
tab1m <- CreateTableOne(strata = "arthritis.type", vars = baselinevars,
                        data = match, test = FALSE)
print(tab1m, smd = TRUE)
```

```
##                            Stratified by arthritis.type
##                             Non-arthritis Yes-arthritis SMD
##   n                         666           666
##   gender = Female (%)       387 (58.1)    412 (61.9)     0.077
##   diabetes = Yes (%)        115 (17.3)    110 (16.5)     0.020
##   smoke = Yes (%)           356 (53.5)    337 (50.6)     0.057
##   age (%)                                                0.091
##      (19.9,50]              122 (18.3)    123 (18.5)
##      (50,70]                358 (53.8)    331 (49.7)
##      (70,81]                186 (27.9)    212 (31.8)
##   annualincome (%)                                       0.039
##      <20k                    131 (19.7)    138 (20.7)
##      20kto54k                279 (41.9)    267 (40.1)
##      55k+                    256 (38.4)    261 (39.2)
##   medical.access = Yes (%)  619 (92.9)    615 (92.3)     0.023
##   blood.pressure = Yes (%)  384 (57.7)    393 (59.0)     0.027
```

```
emp = merge(DT9a, match)
```

Ilustración del PS por regresión de covariables

```
# Primero generamos el modelo para el tratamiento

modelo = glm(PSformula2, data = DT9a, family = binomial())

# Generamos los valores de PS

ps = modelo$fitted

# Añadirlo a la base de datos

DT9a$ps = ps

# Realizar el modelo para la variable que creemos que es
# confusora con variables tratamiento y valor de ps
```

```r
modelocov = glm(heart.attack ~ arthritis.type + ps, data = DT9a, family = binomial())

# Realizar el estudio para ver si ps es significativo con
# la función logistic.display
# install.packages("epiDisplay")

logistic.display(modelocov)
```

```
##
## Logistic regression predicting heart.attack : Yes vs No
##
##                                              crude OR(95%CI)
## arthritis.type: Yes-arthritis vs Non-arthritis 1.99 (1.31,3)
##
## ps (cont. var.)                              65.1 (24.96,169.78)
##
##                                              adj. OR(95%CI)
## arthritis.type: Yes-arthritis vs Non-arthritis 0.89 (0.56,1.42)
##
## ps (cont. var.)                              72.79 (25.29,209.51)
##
##                                              P(Wald's test) P(LR-test)
## arthritis.type: Yes-arthritis vs Non-arthritis 0.627          0.626
##
## ps (cont. var.)                              < 0.001        < 0.001
##
## Log-likelihood = -455.6817
## No. of observations = 3549
## AIC value = 917.3634
```

```r
modelocrudo = glm(heart.attack ~ arthritis.type, data=DT9a, family=binomial())
logistic.display(modelocrudo)
```

```
##
## Logistic regression predicting heart.attack : Yes vs No
##
##                                              OR(95%CI)      P(Wald's test)
## arthritis.type: Yes-arthritis vs Non-arthritis 1.99 (1.31,3)  0.001
##
##                                              P(LR-test)
## arthritis.type: Yes-arthritis vs Non-arthritis 0.002
##
## Log-likelihood = -485.5788
## No. of observations = 3549
## AIC value = 975.1575
```

```r
modelocrudoemparejamiento = glm(heart.attack ~ arthritis.type, data=emp, family=binomial)

logistic.display(modelocrudoemparejamiento)
```

```
##
## Logistic regression predicting heart.attack : Yes vs No
##
##                                              OR(95%CI)         P(Wald's test)
## arthritis.type: Yes-arthritis vs Non-arthritis 0.82 (0.51,1.31)  0.41
```

```
##
##                                                 P(LR-test)
## arthritis.type: Yes-arthritis vs Non-arthritis 0.409
##
## Log-likelihood = -288.3357
## No. of observations = 1333
## AIC value = 580.6715
```

```
modelooriginal = glm(heart.attack ~ arthritis.type + gender +
                       diabetes + smoke + age +
  annualincome + medical.access + blood.pressure,
  data=DT9a, family = binomial())

summary(modelooriginal)
```

```
##
## Call:
## glm(formula = heart.attack ~ arthritis.type + gender + diabetes +
##     smoke + age + annualincome + medical.access + blood.pressure,
##     family = binomial(), data = DT9a)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9638  -0.2527  -0.1267  -0.0747   3.5590
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -4.59701    0.46988  -9.783  < 2e-16 ***
## arthritis.typeYes-arthritis -0.05112    0.23069  -0.222 0.824638
## genderFemale                 -1.16538    0.24450  -4.766 1.87e-06 ***
## diabetesYes                   0.34846    0.23174   1.504 0.132661
## smokeYes                      0.11932    0.21526   0.554 0.579375
## age(50,70]                    1.76265    0.37762   4.668 3.05e-06 ***
## age(70,81]                    2.41192    0.39297   6.138 8.38e-10 ***
## annualincome20kto54k         -0.68853    0.23984  -2.871 0.004094 **
## annualincome55k+             -1.28209    0.27393  -4.680 2.86e-06 ***
## medical.accessYes             0.34509    0.35748   0.965 0.334379
## blood.pressureYes             0.84650    0.23329   3.628 0.000285 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 980.82  on 3548  degrees of freedom
## Residual deviance: 787.16  on 3538  degrees of freedom
## AIC: 809.16
##
## Number of Fisher Scoring iterations: 8
```