

**Profesor:** Héctor Bahamonde, PhD.

**e:** [hector.bahamonde@uoh.cl](mailto:hector.bahamonde@uoh.cl)

**w:** [www.hectorbahamonde.com](http://www.hectorbahamonde.com)

**Curso:** MLE.

**TA:** Gonzalo Barria.

## I. MLE Y CAUSAL INFERENCE: MATCHING, COVARIATE BALANCE, Y EL PROPENSITY SCORE

**Motivación** De acuerdo a Morgan and Winship (2007, p. 33), el *treatment effect* (TE) “naive”  $\delta_i$  para el sujeto  $i$  es,

$$\delta_i = y_i^T - y_i^C \quad (1)$$

donde  $y$  es la variable dependiente bajo un régimen de tratamiento (T) o control (C). Como ya sabemos, este TE es “naive” en el sentido de que es una cantidad de interés no observada: es imposible observar los dos estados para el mismo sujeto  $i$ . O ves uno o ves el otro. Esto es lo que de manera muy conocida Holland (1986, p. 947) ha llamado el *fundamental problem of causal inference*.

Es por esto que nosotros trabajamos con el *average treatment effect* (ATE).

$$\text{ATE} = \frac{1}{n} \sum_{i=1}^n E[Y_i^T - Y_i^C \mid X_i] \quad (2)$$

Como ves, el efecto causal “**average**” es la resta del grupo que fue tratado (T) menos el grupo que no fue tratado (C) (Ho et al. 2007, p. 204).

La lógica experimental indica que en vez de controlar por *covariados* (como se hace en una regresión lineal tipo OLS), controlas por nada (Guo and Fraser 2009, p. 6). Lo que termina “controlando” es la aleatorización del régimen experimental (T o C). **Por qué?** Lo que logra la randomización, de manera más formal, es lo siguiente:

$$i \perp\!\!\!\perp C, T \quad (3)$$

En palabras, la asignación de la unidad  $i$  a uno de los dos regímenes es *independiente* (“ $\perp$ ”) de las características de la unidad  $i$ . **Por qué esto es importante?**

Aunque es claro que todos quisiéramos realizar experimentos para poder calcular el ATE en Equation 2, esto sólo es posible en contextos experimentales. En el resto de las veces (observacional), esto no será posible. Muchas veces los experimentos no están disponibles debido a la imposibilidad de hacer uno (costo, criterios éticos, etc.). **Cómo podemos “simular” un experimento cuando no podemos conseguir uno?** Una técnica posible es el “**matching**”.

**“Covariate Balance”** Lo que la randomización experimental logra es *covariate balance*, es decir, que **TODAS** las características *pre-treatment* sean balanceadas en los dos regímenes experimentales. Es decir, todo lo que observamos de  $i$  (y lo que *no* observamos también) estará distribuido de igual manera (i.e. “balanceado”) en ambos regímenes. El *covariate balance* es fundamental porque hace que las características de todos los  $i$  (genero, ingresos, ideología, etc.) estén distribuidas de igual maneras. Lo importante es que cuando eso ocurre, las diferencias (entre genero, por ejemplo) se cancelarán entre ellas. Lo mismo ocurre con todo el resto de carecterísticas (observables y no-observables). **Qué ventajas trae esto en relación a la regresión?**

Lo que hace el **matching** es encontrar *pares similares* (o “matches”) entre el grupo C y el grupo T. La idea es lograr (**usando datos observacionales**) el *covariate balance* que logra un experimento (sin tener uno).

**Propensity Score Matching** Las técnicas de *matching* trabajan con **distancias** que indican cuán iguales o diferentes son las unidades  $i$ .<sup>1</sup> Es sobre esas distancias que determinamos si una unidad  $i_T$  encuentra su “match”  $i_C$ . La distancia más usada es el “*propensity score*” (PS). **El PS es la probabilidad de  $i$  de ser asignado al régimen T.** Si conocemos estas probabilidades, podemos tratar de imitar un experimento (pero usando datos observacionales), específicamente, en lo que respecta al *covariate balance*. Si conocemos la probabilidad de ser asignado al tratamiento (que es la definición del PS), podemos balancear nuestros datos y **botar** todas las observaciones que no encuentren su *match* (en función de la probabilidad del PS). Al botar observaciones que no tengan pares similares en cuanto al PS estaremos **balanceando nuestros datos**. Y es que de esta manera imitamos un experimento cuando no podemos realizar uno.

---

<sup>1</sup>Hay varias clases de distancias. En esta clase sólo veremos el *propensisty score*.

Más formalmente, lo que nosotros queremos lograr es que la variable dependiente  $y$  sea “ortogonal” (o independiente, i.e.  $\perp$ ) del mecanismo de asignación al tratamiento  $\tau$ —un experimento logra eso de manera natural, no así un diseño observacional donde puede que exista una correlación  $\text{Corr}(y, \tau) > 0$ . El PS ayuda a que, cuando es condicionado en el *propensity score* (o “*balancing score*”)  $b(X)$   $y$  sea ortogonal a  $\tau$ , de manera tal que,

$$y_i \perp \tau_i | x_i \equiv y_i \perp \tau_i | b(X)_i \quad (4)$$

Estadísticamente, lo que ocurre al balancear  $y_i$  en cuanto al PS  $b(X)_i$  es que ambas distribuciones se hacen indistinguibles. **Por qué?** Y eso es justamente lo que buscamos: que ambos grupos T y C sean iguales, excepto en  $\tau_i$ . Piensa en un experimento: ambos grupos (T, C) son iguales (por el efecto de la randomización), **excepto** en que un  $i_T$  es tratado y el otro  $i_C$  no es tratado.

Claramente, todo esto supone que nosotros conocemos (o al menos, podemos construir) el PS. Para hacerlo, debemos estimar una regresión logística. La variable dependiente es el vector  $\tau_{0,1}$  y las variables independientes están compuestas por una matriz  $x$  que contenga características *pre-treatment*. Formalmente, necesitamos estimar,

$$\tau = \mathbf{x}_i \beta + \epsilon_i \quad (5)$$

donde  $\mathbf{x}_i$  es la matriz  $x$  que contiene características *pre-treatment*. Nota que  $\tau \sim \text{Logistic}(\mu, \frac{\sigma^2 \pi^2}{3})$ , y donde  $\sigma^2$  es proporcional a la desviación estándar.

**“Matching as a Preprocessing Tool”** Ho et al. (2007) explica que **matching se debe ocupar como una herramienta para pre-procesar los datos**. Recuerda que *matching*, lo que esencialmente hace, es botar observaciones. *Matching* no es una herramienta de inferencia estadística propiamente tal. En consecuencia, los autores recomiendan hacer *matching* **antes** de estimar cualquier modelo paramétrico. La idea es primero obtener datos quasi-experimentales (vía *matching*) y después estimar modelos OLS o GLM (u otros).

- **Ventajas:**

1. Una base de datos pre-procesada es menos dependiente del modelo. **Por qué es esto importante?**

2. Nos acercamos a un diseño experimental.

- **Desventajas:**

1. El PS es altamente dependiente de los *pre-treatment covariates*  $x$ .
2. El PS “se vende” como no-parametrico, i.e. libre de supuestos distribucionales e inferenciales. **Es?**
3. Matching selecciona en observables. La experimentación en observables y no-observables.
4. Existen muchas distancias: poca claridad de un metodo unico de *matching*. Cacofonía.

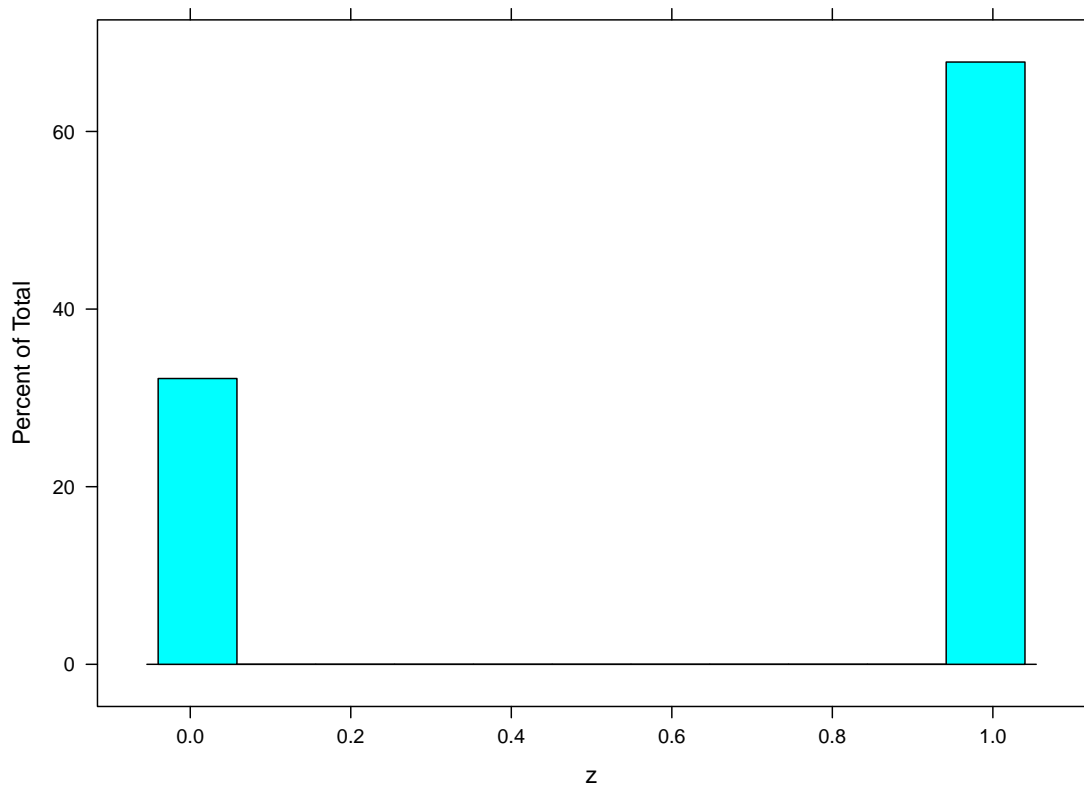
## I. Estimación

Estimemos modelos sin *matching* (“raw”) y con *matching*. Simulemos unos datos.

```
set.seed(2020) # seed
#
e = rnorm(1100, 0) # residuo
x1 = rnorm(1100, -10) # x1
x2 = rnorm(1100, 20) # x2
# mecanismo de asignacion a tratamiento sesgado: depende de x1 y una constante.
t <- as.numeric(1 + 0.02*x1)
pr = as.numeric(1/(1+exp(-t))) # poner t en una funcion logit invertida
z = rbinom(1100,1,pr) # crear z como funcion binomial basado en el vector de prob's
#
y <- as.numeric(1*z + 1*x1 + 2*x2 + e) # y es combinacion lineal de factores
#
dat = as.data.frame(cbind(y, x1, x2,z)) # crear base de datos
```

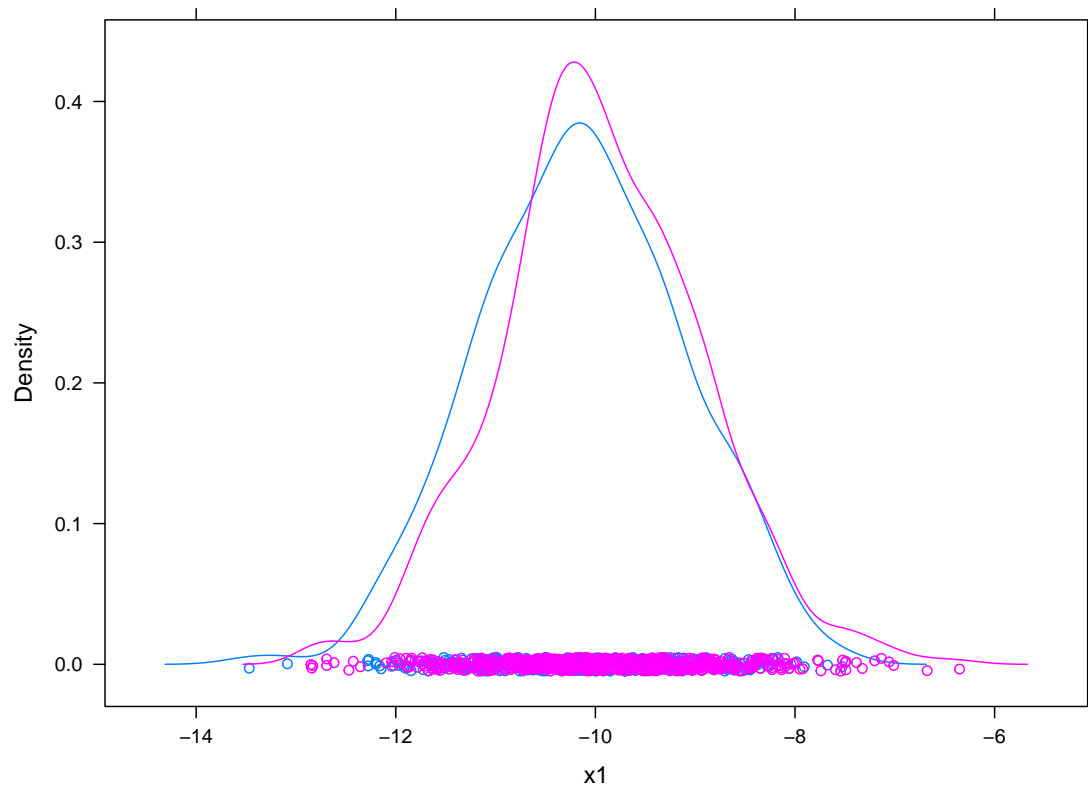
Veamos la distribución de  $z$ .

```
library(lattice)
histogram(z)
```

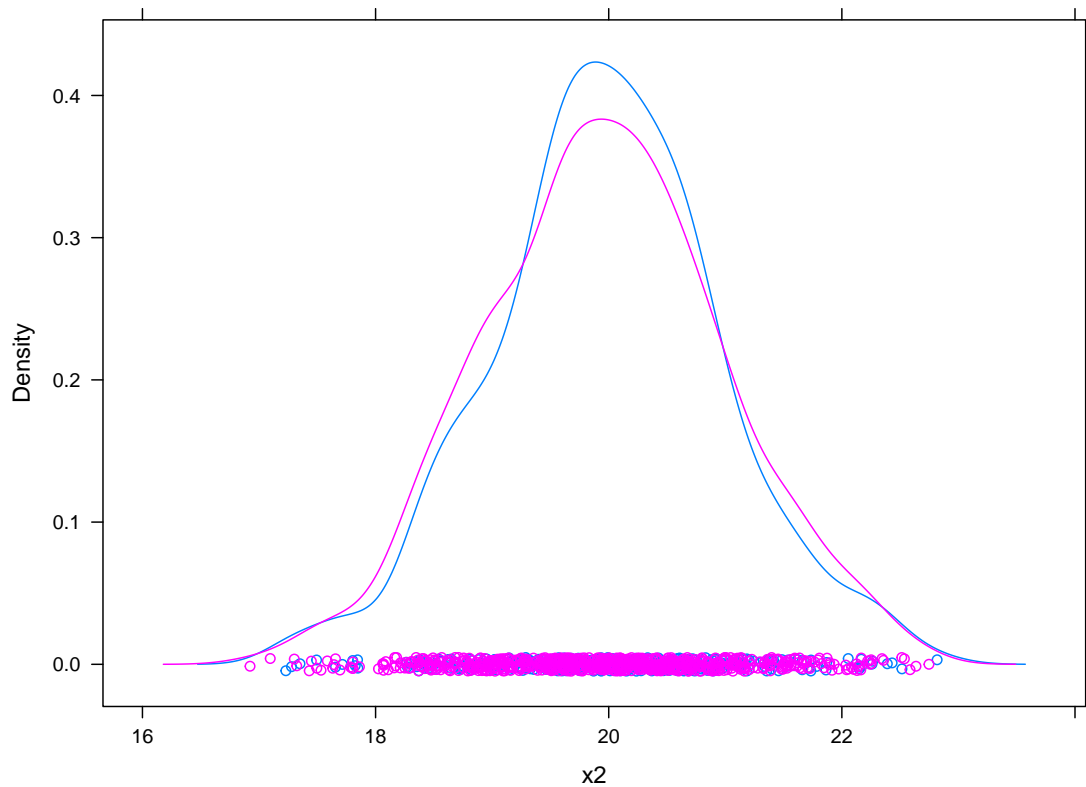


Veamos el *imbalance* de los covariados.

```
library(lattice)
densityplot(x1, groups = z, data = dat)
```



```
densityplot(x2, groups = z, data = dat)
```



Hagamos matching.

```
# install.packages("MatchIt", dependencies = T)
library("MatchIt")
m.out = matchit(z ~ x1 + x2,
data = dat,
method = "nearest", # nearest: "nearest neighbor"
discard = "both", # botar obs en ambas (T y C)
distance = "logit",
caliper = .1 # ventana de comparacion entre T y C
)
```

Veamos si hubo mejoras.

```
summary(m.out) # inspeccionar balance y obs botadas

##
## Call:
## matchit(formula = z ~ x1 + x2, data = dat, method = "nearest",
## distance = "logit", discard = "both", caliper = 0.1)
##
## Summary of Balance for All Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance      0.6807      0.6729      0.1891      0.9796      0.0508
## x1           -9.9522     -10.1411      0.1872      1.0032      0.0498
## x2            19.9585      19.9825     -0.0233      1.1028      0.0160
##           eCDF Max
## distance      0.1001
## x1            0.0967
## x2            0.0514
##
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance      0.6781      0.6746      0.0845      1.0078      0.0261
## x1           -10.0172     -10.1043      0.0864      1.0074      0.0255
## x2            19.9823      19.9693      0.0125      1.1600      0.0170
##           eCDF Max Std. Pair Dist.
## distance      0.0520      0.0848
## x1            0.0694      0.1553
## x2            0.0405      1.1098
##
## Percent Balance Improvement:
##           Std. Mean Diff. Var. Ratio eCDF Mean eCDF Max
## distance      55.3      62.2      48.6      48.1
```

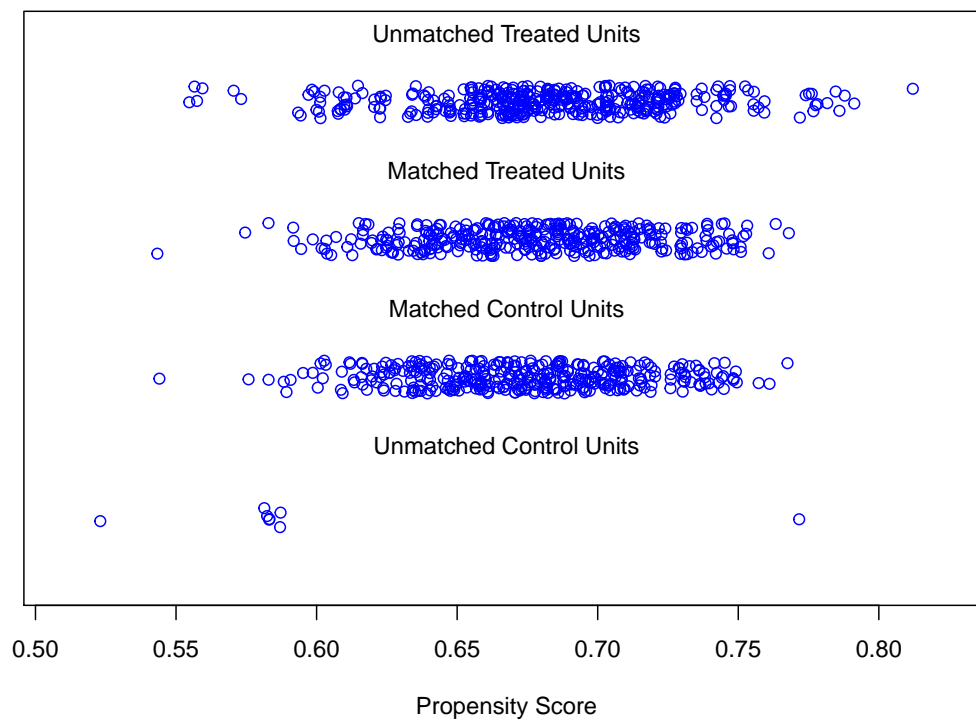


```
## x1          53.9    -130.4    48.7    28.3
## x2          46.1    -51.6    -6.8    21.3
##
## Sample Sizes:
##           Control Treated
## All          354     746
## Matched      346     346
## Unmatched     7     387
## Discarded     1      13
```

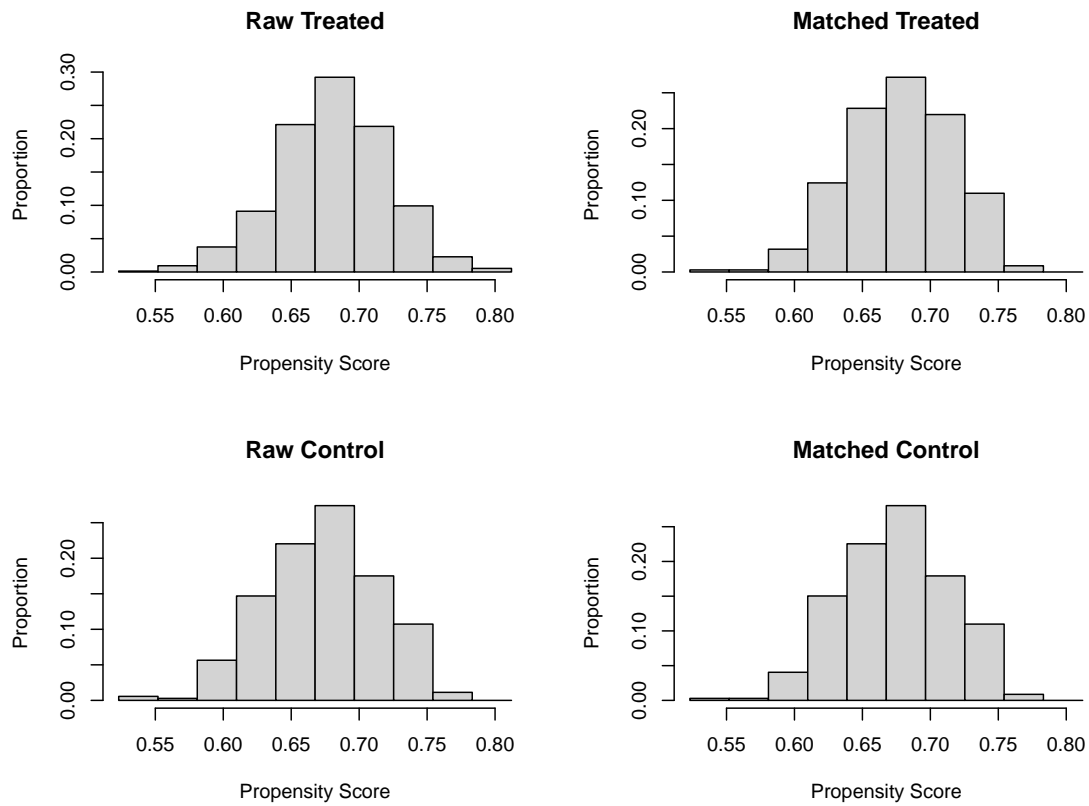
Sigamos inspeccionando de manera visual.

```
plot(m.out, type = "jitter", interactive = FALSE, col = "blue")
```

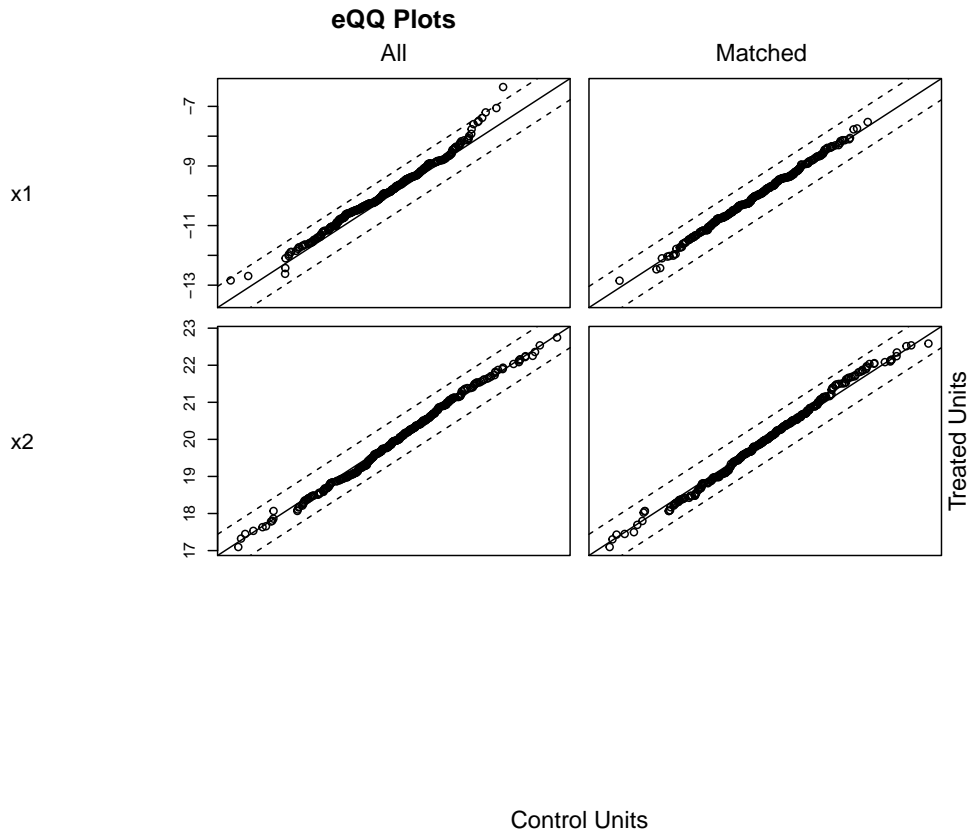
### Distribution of Propensity Scores



```
plot(m.out, type = "hist")
```



```
plot(m.out, type = "qq", interactive = FALSE)
```



OK. Ahora, botemos las observaciones creando una base de datos nueva `m.data`.

```
m.data = match.data(m.out, distance = "prop.score", drop.unmatched = T) # construir DF
```

Ahora, estimemos dos modelos, uno con todos los datos (“raw”) y otro con los datos *matcheados* (“matched”).

```
raw.m = lm(y ~ z + x1 + x2, data = dat) # OLS (Raw)
matched.m = lm(y ~ z + x1 + x2, data = m.data) # OLS (Matched)
library(texreg)
texreg(list(raw.m, matched.m))
```

	Model 1	Model 2
(Intercept)	−0.56 (0.68)	−1.05 (0.91)
z	1.06*** (0.07)	0.98*** (0.08)
x1	0.99*** (0.03)	0.96*** (0.04)
x2	2.02*** (0.03)	2.03*** (0.04)
R <sup>2</sup>	0.84	0.84
Adj. R <sup>2</sup>	0.84	0.84
Num. obs.	1100	692

\*\*\*  $p < 0.001$ ; \*\*  $p < 0.01$ ; \*  $p < 0.05$

**Table 1:** *Statistical models*

```
knitr::purl('Matching.Rnw')

## Error in parse_block(g[-1], g[1], params.src, markdown_mode): Duplicate chunk label
## 'setup', which has been used for the chunk:
## if (!require("pacman")) install.packages("pacman"); library(pacman)
## p_load(knitr)
## set.seed(2020)
## options(scipen=9999999)

Stangle('Matching.Rnw')

## Writing to file Matching.R

## Error in match.arg(options$results, c("verbatim", "tex", "hide")): 'arg' should
## be one of "verbatim", "tex", "hide"
```

## REFERENCES

- Guo, Shenyang and Mark Fraser (2009). *Propensity Score Analysis: Statistical Methods and Applications*. SAGE Publications.
- Ho, Daniel et al. (Dec. 2007). “Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference.” In: *Political Analysis* 15.3, pp. 199–236.
- Holland, Paul (1986). “Statistics and Causal Inference: Rejoinder.” In: *Journal of the American Statistical Association* 81.396, pp. 968–970.
- Morgan, Stephen and Christopher Winship (July 2007). *Counterfactuals and Causal Inference: Methods and Principles for Social Research*. Analytical Methods for Social Research. Cambridge University Press.