Targeted Marketing

Isidoro Garcia

2021

Overview

Los contratan como data scientists para una empresa que vende electrodomesticos. La empresa lanzó un experimento de control aleatorio via un mail en donde se envió un catalogo de los productos al grupo de tratamiento `mailing_indicator`.

Tu objetivo es estimar el impacto del envío sobre el gasto incremental:

$$\tau_i = \mathbb{E}[Y_i(1) - Y_i(0)|\boldsymbol{x}_i],$$

En particular, queremos estimar el impacto de enviar el catalogo a nivel de cliente. Para ello, pondremos a competir algunos de los modelo de Causal Machine Learning que hemos aprendido en clase:

- Double Debiased Machine Learning
- Causal Forests

Adicionalmente, desarrollen una estrategia de focalización con base en los resultados de tu modelo. Elabora sobre la lógica económica (i.e. identifica los Beneficios y Costos Marginales de enviar la campaña). Finalmente, corrobora la validez externa de la estrategia usando datos de un año. Esto nos dará un termómetro de la utilidad del modelo para campañas posteriores.

Paso 1: Estimación y predicción the Conditional Average Treatment Effects (CATE)

Carguemos los datos de 2015

```
library(tidyverse)
library(data.table)
library(gamlr)
library(grf)
library(xgboost)
library(ranger)
library(RCT)
library(lfe)
library(stargazer)
library(knitr)
```

```
load("Bases input/Customer-Development-2015.RData")
```

Dividimos la base en entrenamiento y validacion. Usamos un seed fijo para replicabilidad.

```
set.seed(1990)
crm<-
  crm %>%
  mutate(training_sample = rbinom(n = nrow(crm), 1, 0.7))
```

Data cleaning

1. Haz una primera revisión de la base. Cuantas variables tienen `NA`

```
missings<-map_dbl(crm %>% select_all(), ~100*sum(is.na(.))/nrow(crm))
kable(missings)
```

	x
customer_type_L	0
customer_type_C	0
clicks_product_type_504_3m	0
clicks_product_type_112_6m	0
clicks_product_type_301_1m	0
clicks_product_type_001_12m	0
clicks_product_type_201_1m	0
clicks_product_type_201_6m	0
web_activity_3m	0
clickthrough_1m	0
emailview_1m	0
orders_online_1yr	0
online_customer	0
orders_season_E	0
orders_mail_dept_h_1yr	0
spend_d_h_1yr	0
spend_online_attributed_target	0
orders_online_attributed_target	0
dollars_season_C	0
spend_season_C	0
spend_e	0
spend_z1	0
spend_period_1b	0
spend_period_2b	0
spend_h_3yr	0
spend_g_3yr	0
last_years_since	0
spend_instore_o	0
spend_instore_o_3yr	0
spend_instore_h_3yr	0
spend_instore_t	0
spend_online_sh	0
spend_online_o_1yr	0
spend_online_n_3yr	0
spend_online_o_3yr	0
spend_online_a_1yr	0
spend_online_a_3yr	0
orders_online_t_3yr	0
spend_online_sg_1yr	0
spend_online_g_1yr	0
orders_online_s_1yr	0
customer_type_7	0
customer_type_A	0
orders_attributed_mail_type_B	0
spend_attributed_mail_type_B	0
spend_attributed_mail_type_C	0
mean_spend_attributed_mail_type_C	0

	x
mean_spend_attributed_mail_type_A	0
clicks_product_type_104_2yr	0
clicks_product_type_312_3yr	0
orders_e	0
orders_mail_dept_c_1yr	0
spend_d_s_1yr	0
spend_d_k_1yr	0
spend_a_1yr	0
spend_h_1yr	0
spend_direct_1yr	0
acquisition_months_since	0
spend_online_c	0
spend_online_a_6m	0
orders_online_o	0
orders_online_c_1yr	0
spend_online_b_1yr	0
customer_type_2	0
acquisition_days_since	0
in_database_months	0
customer_income	0
orders_attributed_mail_type_A	0
spend_attributed_mail_type_A	0
clicks_product_type_312_3m	0
clicks_product_type_301_2yr	0
clickthrough_6m	0
emails_days_2yr	0
emails_3m	0
emailreceived_months_since	0
orders_3yr	0
orders_mail_dept_d_1yr	0
spent_q	0
spend_instore_q	0
spend_online_s	0
clicks_product_type_112_3yr	0
spend_instore_n_3yr	0
spend_instore_p	0
spend_instore_p_1yr	0
orders_online_n_1yr	0
spend_instore_a_yr	0
orders_attributed_mail_type_C	0
clickthrough_3yr	0
spend_instore_g_1yr	0
spend_period_3b	0
spend_instore_a	0
spend_direct_g	0
emailview_24m	0
spend_online_h_3yr	0
emailview_months_since	0
orders_instore_c	0
customer_type_3	0
emails_1yr	0
clicks_product_type_001_1m	0

	x
clickthrough_3m	0
orders_d_1yr	0
orders_h	0
clicks_product_type_104	0
clicks_product_type_502_3m	0
web_activity_1m	0
orders_instore_m	0
spend_notz_1yr	0
orders_online_sh	0
orders_instore_n	0
clicks_product_type_502_1yr	0
orders_hm	0
emails_days_1yr	0
emailview	0
spend_m_1yr	0
clicks_product_type_301_3m	0
orders_instore_a_yr	0
clicks_product_type_001_6m	0
clickthrough_months_since	0
orders_online_h_1yr	0
orders_instore_h_3yr	0
spend_period_1a	0
orders_total	0
spend_online_s_3yr	0
spend_M_3yr	0
orders_c	0
spend_l	0
spend_instore_n	0
orders_z	0
web_activity_24m	0
spend_instore_q_3yr	0
clicks_product_type_504	0
emailview_6m	0
buy_instore_days_since	0
spend_h	0
clicks_product_type_801_3yr	0
spend_nott_1yr	0
orders_online_s_3yr	0
spend_instore_h_1yr	0
spend_online_b_3yr	0
orders_instore_s_3yr	0
orders_instore_r_3yr	0
clicks_product_type_502_1m	0
emailview_3m	0
mean_spend_attributed_mail_type_B	0
clicks_product_type_201_3yr	0
store_trips	0
spend_online_t_1yr	0
orders_online_o_1yr	0
spend_online_k	0
spend_online_s_1yr	0
clicks_product_type_503	0

	x
orders_instore_t_3yr	0
outcome_spend	0
mailing_indicator	0
customer_id	0
training_sample	0

2. Muestra la matriz de correlación entre variables. Muestra los pares de variables que tienen más de 95% de correlación. Remueve una de cada par multicolineal.

```
cor_matrix <- cor(crm %>% select(-customer_id))

cor_matrix[upper.tri(cor_matrix, diag = TRUE)] = NA

cor_tibble<-tibble(row = rep(rownames(cor_matrix), ncol(cor_matrix)),
                   col = rep(colnames(cor_matrix), each = ncol(cor_matrix)),
                   cor = as.vector(cor_matrix))

cor_tibble<-
  cor_tibble %>%
  filter(!is.na(cor))

large_cor_tibble<-
  cor_tibble %>%
  filter(abs(cor)>=0.95)

kable(large_cor_tibble, digits = 3)
```

row	col	cor
customer_type_3	online_customer	-0.958
orders_online_attributed_target	spend_online_attributed_target	0.965
acquisition_days_since	acquisition_months_since	1.000
in_database_months	acquisition_months_since	1.000
in_database_months	acquisition_days_since	1.000
emails_days_1yr	emails_days_2yr	0.960
emailview_3m	emailview_6m	0.950

```
# Quitando las variables de col
crm<-
  crm %>%
  select(-all_of(large_cor_tibble$col))
```

3 (2 pts). Corroba que la asignación tratamiento fue aleatoria mediante revisión del balance. Realiza las pruebas balance T y F. Cuántas variables salen desbalanceadas? Que muestra esto sobre la asignación de tratamiento?

```
balance_t<-balance_table(crm %>% select(-customer_id, -outcome_spend), "mailing_indicator")
kable(balance_t, digits = 2)
```

variables1	Media_control1	Media_trat1	p_value1
buy_instore_days_since	1038.48	1037.57	0.81
clicks_product_type_001_12m	1.56	1.64	0.14

variables1	Media_control1	Media_trat1	p_value1
clicks_product_type_001_1m	0.06	0.07	0.22
clicks_product_type_001_6m	0.66	0.67	0.46
clicks_product_type_104	1.20	1.24	0.09
clicks_product_type_104_2yr	0.20	0.21	0.12
clicks_product_type_112_3yr	3.81	3.89	0.24
clicks_product_type_112_6m	1.68	1.70	0.55
clicks_product_type_201_1m	0.15	0.16	0.05
clicks_product_type_201_3yr	5.44	5.54	0.20
clicks_product_type_201_6m	2.31	2.34	0.48
clicks_product_type_301_1m	0.06	0.06	0.12
clicks_product_type_301_2yr	1.69	1.70	0.81
clicks_product_type_301_3m	0.24	0.24	0.93
clicks_product_type_312_3m	0.55	0.56	0.89
clicks_product_type_312_3yr	4.99	5.01	0.81
clicks_product_type_502_1m	0.56	0.55	0.73
clicks_product_type_502_1yr	10.59	10.63	0.77
clicks_product_type_502_3m	2.11	2.11	0.92
clicks_product_type_503	79.53	80.05	0.68
clicks_product_type_504	43.10	43.59	0.36
clicks_product_type_504_3m	2.53	2.55	0.79
clicks_product_type_801_3yr	2.85	2.86	0.91
clickthrough_1m	0.13	0.13	0.30
clickthrough_3m	0.46	0.47	0.13
clickthrough_3yr	3.79	3.85	0.20
clickthrough_6m	1.05	1.07	0.19
clickthrough_months_since	13.86	13.88	0.85
customer_income	73813.35	73911.74	0.61
customer_type_2	0.23	0.23	0.20
customer_type_3	0.17	0.18	0.24
customer_type_7	0.02	0.02	0.70
customer_type_A	0.25	0.25	0.77
customer_type_C	0.50	0.50	0.31
customer_type_L	0.17	0.18	0.04
dollars_season_C	474.48	483.51	0.03
emailreceived_months_since	9.09	9.15	0.50
emails_1yr	104.53	104.81	0.43
emails_3m	30.27	30.34	0.47
emails_days_1yr	119.56	119.92	0.44
emailview	118.28	118.44	0.82
emailview_1m	3.21	3.20	0.67
emailview_24m	48.87	48.88	0.97
emailview_3m	7.54	7.52	0.72
emailview_months_since	12.66	12.74	0.33
in_database_months	241.10	241.96	0.18
last_years_since	0.91	0.91	0.80
mean_spend_attributed_mail_type_A	37.72	37.74	0.93
mean_spend_attributed_mail_type_B	46.23	46.54	0.22
mean_spend_attributed_mail_type_C	83.28	83.72	0.10
orders_3yr	5.66	5.75	0.01
orders_attributed_mail_type_A	1.18	1.19	0.12
orders_attributed_mail_type_B	1.67	1.69	0.12
orders_attributed_mail_type_C	4.73	4.80	0.01

variables1	Media_control1	Media_trat1	p_value1
orders_c	1.70	1.74	0.03
orders_d_1yr	0.78	0.80	0.03
orders_e	0.62	0.63	0.14
orders_h	8.75	8.83	0.33
orders_hm	3.22	3.27	0.06
orders_instore_a_yr	1.02	1.02	0.92
orders_instore_c	0.43	0.43	0.50
orders_instore_h_3yr	0.27	0.27	0.28
orders_instore_m	0.77	0.77	0.98
orders_instore_n	2.37	2.37	0.83
orders_instore_r_3yr	0.08	0.08	0.91
orders_instore_s_3yr	1.46	1.46	0.68
orders_instore_t_3yr	0.23	0.23	0.80
orders_mail_dept_c_1yr	0.53	0.53	0.34
orders_mail_dept_d_1yr	0.73	0.74	0.25
orders_mail_dept_h_1yr	0.24	0.24	0.59
orders_online_1yr	1.41	1.43	0.02
orders_online_attributed_target	15.00	15.22	0.09
orders_online_c_1yr	0.08	0.08	0.30
orders_online_h_1yr	0.69	0.69	0.57
orders_online_n_1yr	0.49	0.49	0.62
orders_online_o	3.90	3.95	0.26
orders_online_o_1yr	0.53	0.54	0.20
orders_online_s_1yr	0.16	0.16	0.03
orders_online_s_3yr	0.45	0.46	0.15
orders_online_sh	0.55	0.55	0.80
orders_online_t_3yr	0.23	0.23	0.37
orders_season_E	8.25	8.36	0.02
orders_total	34.24	34.65	0.04
orders_z	0.07	0.07	0.55
spend_a_1yr	25.74	26.74	0.05
spend_attributed_mail_type_A	101.36	102.13	0.37
spend_attributed_mail_type_B	147.33	148.48	0.34
spend_attributed_mail_type_C	455.40	462.69	0.01
spend_d_h_1yr	96.60	97.91	0.17
spend_d_k_1yr	8.14	8.44	0.14
spend_d_s_1yr	15.71	16.45	0.03
spend_direct_1yr	75.82	76.42	0.52
spend_direct_g	133.63	135.82	0.11
spend_e	68.87	70.18	0.10
spend_g_3yr	144.27	146.24	0.29
spend_h	436.65	442.53	0.11
spend_h_1yr	49.62	49.65	0.97
spend_h_3yr	123.69	124.74	0.36
spend_instore_a	914.53	914.03	0.93
spend_instore_a_yr	111.98	112.13	0.84
spend_instore_g_1yr	1.14	1.14	0.91
spend_instore_h_1yr	6.57	6.51	0.60
spend_instore_h_3yr	17.47	17.51	0.89
spend_instore_n	83.74	84.26	0.39
spend_instore_n_3yr	45.36	45.93	0.12
spend_instore_o	180.29	180.53	0.86

variables1	Media_control1	Media_trat1	p_value1
spend_instore_o_3yr	60.49	61.06	0.28
spend_instore_p	14.61	14.38	0.09
spend_instore_p_1yr	2.00	1.90	0.02
spend_instore_q	5.75	5.89	0.27
spend_instore_q_3yr	4.24	4.32	0.40
spend_instore_t	47.66	47.49	0.64
spend_l	45.40	46.10	0.21
spend_m_1yr	8.23	8.00	0.24
spend_M_3yr	171.17	172.51	0.31
spend_nott_1yr	13.22	13.03	0.40
spend_notz_1yr	234.63	237.71	0.04
spend_online_a_1yr	169.86	172.90	0.01
spend_online_a_3yr	385.69	392.13	0.01
spend_online_a_6m	109.42	110.81	0.06
spend_online_b_1yr	20.12	21.20	0.01
spend_online_b_3yr	51.43	54.08	0.00
spend_online_c	32.83	33.15	0.47
spend_online_g_1yr	66.02	67.35	0.04
spend_online_h_3yr	155.81	155.78	0.98
spend_online_k	69.50	69.91	0.71
spend_online_n_3yr	93.84	93.87	0.98
spend_online_o_1yr	43.95	44.86	0.09
spend_online_o_3yr	104.86	106.62	0.17
spend_online_s	81.87	82.62	0.46
spend_online_s_1yr	12.94	13.44	0.05
spend_online_s_3yr	30.68	30.88	0.68
spend_online_sg_1yr	2.36	2.42	0.63
spend_online_sh	26.31	26.34	0.92
spend_online_t_1yr	10.20	10.05	0.37
spend_period_1a	132.78	133.98	0.16
spend_period_1b	83.41	85.62	0.01
spend_period_2b	62.26	63.12	0.33
spend_period_3b	97.17	98.33	0.20
spend_season_C	1037.89	1051.64	0.04
spend_z1	282.19	286.48	0.06
spent_q	9.67	9.79	0.81
store_trips	11.04	11.00	0.49
training_sample	0.70	0.70	0.28
web_activity_1m	3.43	3.47	0.57
web_activity_24m	81.16	82.31	0.24
web_activity_3m	21.78	22.01	0.49

```
table(balance_t$p_value1<0.05)
```

```
FALSE  TRUE
  123    24
```

```
balance_f<-balance_regression(crm %>% select(-customer_id, -outcome_spend), "mailing_indicator")
kable(balance_f$F_test, digits = 3)
```

estadistico	Msj1
F-statistic	0.983
k	147.000
n-k-1	249852.000
F_critical	0.816
p_value	0.543
R cuadrada	0.001

4. Realize un ajuste de False Discovery Rate al 10%. Cuántas variables salen desbalanceadas ahora?

```
# Creamos el ranking de los p-values
balance_t<-
  balance_t %>%
  arrange(p_value1) %>%
  mutate(ranking = row_number())


balance_t<-
  balance_t %>%
  mutate(corte_fdr = 0.1*ranking/nrow(balance_t),
         sig_fdr = if_else(p_value1<=corte_fdr, 'Significativa', 'No significativa'))


table(balance_t$sig_fdr)
```

```
No significativa
             147
```

Estimación de impacto de tratamiento (ATE)

5 (2pts). Estima el impacto promedio de enviar el catalogo vía email. Estima el impacto sin controles y luego agregar dos estimaciones de robustez: 1) Agregando variables que salieron significativas y 2) Agregando variables que salieron significativas con el FDR. Interpreta los resultados

```
# Estimación sin  controles
itt_sin_controles<-felm(outcome_spend ~ mailing_indicator | 0 |0 | 0, data = crm)

# Estimación con controles 25
controles<-str_c(balance_t %>% filter(p_value1<0.05) %>% select(variables1) %>% pull(), collapse = "+")
formula_y<-str_c("outcome_spend~mailing_indicator+",controles, " | 0 | 0 |0")
itt_controles<-felm(as.formula(formula_y), data = crm)

stargazer(itt_sin_controles, itt_controles)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, May 09, 2021 - 22:33:21

```
rm(balance_t, balance_f, itt_controles, itt_sin_controles,
   cor_matrix, cor_tibble, large_cor_tibble)
```

Estimación de efectos heterogeneos Usaremos el training sample para estimar el Conditional Average Treatment Effect de enviar el catalogo sobre el gasto en dólares. Estimaremos dos tipos de modelos (si agregan otro es bienvenido):

(a) Double Debiased LASSO

Table 5:

	Dependent variable:	
	outcome_spend	formula_y
	(1)	(2)
mailing_indicator	2.922***	2.762***
	(0.190)	(0.181)
spend_online_b_3yr		−0.002**
		(0.001)
spend_online_b_1yr		0.003
		(0.002)
spend_online_a_1yr		−0.003***
		(0.001)
spend_attributed_mail_type_C		0.002***
		(0.0004)
orders_attributed_mail_type_C		−0.100**
		(0.042)
orders_3yr		−0.004
		(0.029)
spend_period_1b		0.005***
		(0.001)
spend_online_a_3yr		0.012***
		(0.0004)
orders_season_E		−0.108***
		(0.027)
spend_instore_p_1yr		0.010
		(0.010)
orders_online_1yr		0.795***
		(0.089)
orders_d_1yr		−0.311***
		(0.089)
dollars_season_C		0.003***
		(0.0001)
spend_d_s_1yr		−0.013***
		(0.002)
orders_c		0.010
		(0.028)
orders_online_s_1yr		0.598**
		(0.250)
spend_notz_1yr		0.010***
		(0.001)

(b) Causal Forests

Separa la base de entrenamiento de la de validación

```
crm_training<-
  crm %>%
  filter(training_sample == 1)
crm_validation <-
  crm %>%
  filter(training_sample == 0)

rm(crm)
```

####Double Debiased LASSO

6 (3pts). Estima un Double Debiased LASSO. Asegurate de mostrar el código. (Tip: recuerda que necesitas guardar el LASSO de cada K para poder usarlo en la base de validación)

```
# X's
X<-
  crm_training %>%
  select(-c(customer_id, outcome_spend, training_sample,mailing_indicator))

X<-sparse.model.matrix(~. + 0, data = X)

outcome_spend<-crm_training$outcome_spend
treat<-crm_training$mailing_indicator

k<-treatment_assign(data = crm_training,
                    share_control = 0.2, n_t = 4,
                    strata_varlist = "customer_id",
                    missfits = "global", seed = 1900,
                    key = "customer_id")
```

```
Warning: Unknown or uninitialised column: `treat`.

Warning: Unknown or uninitialised column: `treat`.
```

```
k<-k$data
k<-
  k %>%
  mutate(k = treat + 1)

k<- k %>% ungroup()
k<-k$k

########################
# Cross-fitting
########################
modelo<-map(1:5,
                function(a) {
                  treat_fit <-gamlr(x = X[k!=a, , drop= F], y = treat[k !=a],family="binomial")

                  spend_fit <-gamlr(x = X[k!=a, , drop= F], y = outcome_spend[k !=a])

                  modelos<-list("treat_fit" = treat_fit, "spend_fit" = spend_fit)
```

```
                })

names(modelo)<-str_c("k=", seq(1,5))

scores<-map_dfr(1:5,
                function(a) {
                   treat_hat<-as.numeric(predict(modelo[[a]]$treat_fit,
                                                 newdata = X[k==a, , drop= F],
                                                 type = "response"))

                   spend_hat<-as.numeric(predict(modelo[[a]]$spend_fit,
                                                 newdata = X[k==a, , drop= F],
                                                 type = "response"))

                   treat_resid <- treat[k==a] - treat_hat

                   spend_resid <- outcome_spend[k==a]- spend_hat

                   scores<-bind_cols("treat_hat" = treat_hat, "spend_hat"= spend_hat,
                                     "treat_resid"= treat_resid, "spend_resid" = spend_resid)




                })

scores<-bind_cols(scores, "outcome_spend" = outcome_spend, "treat" = treat)

save(modelo, file = "Modelos/ddml_lasso.Rdata")
save(scores, file = 'Modelos/scores_ddml.Rdata')
```

7 (2pts). Cuál es el impacto de tratamiento promedio? Estimalo de dos maneras: 1) `spend_resid~treat_hat + treat` y 2) `spend~treat_resid`. Sale lo mismo? Justifica tu respuesta

```
mailing_ate<-lm(spend_resid ~ treat_hat + treat, data = scores)

mailing_ate2<-lm(spend_resid ~ treat_resid, data = scores)

stargazer(mailing_ate, mailing_ate2, title = "ATE")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, May 09, 2021 - 22:33:22

8 (3pts). Cuáles son las variables más importantes para las nuisance functions $T_i = g(X_i)+v_i$ y $y_i = m(X_i)+\epsilon_i$? (Tip: toma las variables que tengan $\beta \neq 0$ en cada k y haz un `inner_join`. De ahí muestra el promedio de los coeficientes) Interpreta la función $g(X_i)$, porque sale así?

```
coeficientes_m<-map(modelo,
                    function(x) {



                      coeficientes<-as.matrix(coef(x[["spend_fit"]]))
```

Table 6: ATE

	Dependent variable:	
	spend_resid	
	(1)	(2)
treat_hat	125.532*	
	(70.764)	
treat	2.191***	
	(0.216)	
treat_resid		2.721***
		(0.216)
Constant	−85.516*	−0.002
	(47.379)	(0.101)
Observations	174,882	174,882
R^2	0.001	0.001
Adjusted R^2	0.001	0.001
Residual Std. Error	42.400 (df = 174879)	42.393 (df = 174880)
F Statistic	53.191*** (df = 2; 174879)	159.455*** (df = 1; 174880)
Note:		*p<0.1; **p<0.05; ***p<0.01

```
                      coeficientes<-tibble(variable = rownames(coeficientes),
                      coeficiente = coeficientes[,1])

                      coeficientes<-coeficientes %>% filter(coeficiente !=0)
                      })

coeficientes_g<-map(modelo,
                    function(x) {

                      coeficientes<-as.matrix(coef(x[["treat_fit"]]))

                      coeficientes<-tibble(variable = rownames(coeficientes),
                      coeficiente = coeficientes[,1])

                      coeficientes<-coeficientes %>% filter(coeficiente !=0)
                      })


coeficientes_m<-reduce(coeficientes_m, inner_join, by = "variable")
coeficientes_g<-reduce(coeficientes_g, inner_join, by = "variable")

coeficientes_g<-
  coeficientes_g %>%
  rowwise() %>%
  mutate(coef_final= mean(coeficiente, coeficiente.x, coeficiente.y, coeficiente.x.x, coeficiente.y.y))
  select(variable, coef_final)
```

```r
coeficientes_m<-
  coeficientes_m %>%
  rowwise() %>%
  mutate(coef_final= mean(coeficiente, coeficiente.x, coeficiente.y, coeficiente.x.x, coeficiente.y.y))
  select(variable, coef_final)


kable(
   coeficientes_g %>%
   arrange(desc(abs(coef_final))))
```

variable	coef_final
intercept	0.7035572

```r
kable(
   coeficientes_m %>%
   arrange(desc(abs(coef_final))))
```

variable	coef_final
customer_type_7	-3.6699266
customer_type_3	-2.8568291
last_years_since	-2.4352413
customer_type_2	-1.3830254
intercept	1.1919275
customer_type_L	1.1056087
orders_mail_dept_h_1yr	0.5739799
clicks_product_type_301_1m	0.5325725
customer_type_C	-0.4065591
clicks_product_type_201_1m	0.3651084
orders_instore_r_3yr	-0.3622525
clickthrough_1m	0.2793217
orders_online_1yr	0.2086039
orders_online_s_1yr	0.2039464
orders_attributed_mail_type_B	0.2002928
orders_season_E	-0.1304265
orders_online_attributed_target	0.1043604
emailview_1m	0.0724601
orders_online_o	-0.0690982
orders_3yr	-0.0478465
clicks_product_type_104	0.0392184
clicks_product_type_201_6m	0.0250326
clicks_product_type_504_3m	0.0178776
spend_online_o_1yr	0.0102530
spend_h_1yr	0.0098243
spend_online_a_3yr	0.0080735
spend_d_k_1yr	0.0075619
spend_online_o_3yr	0.0074010
spend_online_sh	-0.0053638
spend_d_h_1yr	0.0048616
spend_period_2b	0.0040766
web_activity_3m	0.0040232

variable	coef_final
spend_instore_o_3yr	0.0038695
spend_instore_t	0.0033145
spend_online_b_1yr	-0.0027581
spend_attributed_mail_type_B	0.0025801
clicks_product_type_504	-0.0025779
spend_period_1b	0.0019251
spend_h_3yr	0.0018861
in_database_months	0.0014334
spend_online_a_1yr	0.0014059
spend_attributed_mail_type_C	0.0013058
dollars_season_C	0.0012735

9 (3pts). Ahora corre un DDML LASSO para encontrar los efectos a nivel cliente (Tip: interactúa todas las variables con `treat_resid`. Muestra el código. Qué varaibles salen significativas?

```
crm_training<-
  bind_cols(crm_training, scores %>% select(treat_resid))

X<-
  crm_training %>%
  select(-c(customer_id, outcome_spend, training_sample, mailing_indicator))

X<-sparse.model.matrix(~. + 0+ . *treat_resid, data = X)
outcome_spend<-crm_training$outcome_spend

ddml_hte<-gamlr(x = X, y = outcome_spend, free = "treat_resid")
```

```
[1] 147
```

```
save(ddml_hte, file = 'Modelos/ddml_hte.Rdata')

coeficientes<-as.matrix(coef(ddml_hte))

coeficientes<-tibble(variable = rownames(coeficientes),
                     coeficiente = coeficientes[,1])

coeficientes<-coeficientes %>% filter(coeficiente !=0)

kable(
   coeficientes<-
   coeficientes %>%
   arrange(desc(abs(coeficiente))))
```

variable	coeficiente
customer_type_7	-4.3580077
customer_type_3	-2.5965055
last_years_since	-2.3616165
intercept	1.0929968
customer_type_2	-0.9353825
customer_type_C	-0.8377130
customer_type_L	0.8010180
customer_type_L:treat_resid	0.6861899

variable	coeficiente
orders_online_s_1yr	0.6413995
orders_mail_dept_h_1yr	0.6169066
clicks_product_type_301_1m:treat_resid	-0.5944294
orders_instore_r_3yr	-0.4521413
clickthrough_1m	0.4339071
clicks_product_type_301_1m	0.3954026
orders_instore_r_3yr:treat_resid	-0.3582495
clicks_product_type_201_1m	0.2956035
orders_online_sh:treat_resid	0.2855924
customer_type_3:treat_resid	0.2824834
orders_mail_dept_d_1yr:treat_resid	0.2654595
orders_attributed_mail_type_B	0.1975397
orders_mail_dept_c_1yr:treat_resid	0.1812534
orders_online_1yr	0.1526958
orders_season_E	-0.1358601
orders_online_attributed_target	0.1186697
orders_e:treat_resid	0.1093049
orders_online_o	-0.1032443
treat_resid	-0.0801499
orders_instore_a_yr	-0.0789305
orders_online_t_3yr	0.0707166
emailview_1m	0.0681512
clicks_product_type_201_6m	0.0566271
clicks_product_type_312_3m	-0.0562563
clicks_product_type_001_12m:treat_resid	0.0542009
spend_online_sg_1yr:treat_resid	-0.0536279
web_activity_1m:treat_resid	-0.0337076
clicks_product_type_001_1m	-0.0279926
clicks_product_type_504_3m	0.0254181
clicks_product_type_801_3yr:treat_resid	0.0234474
clicks_product_type_104	0.0232334
clicks_product_type_301_2yr:treat_resid	0.0192692
orders_3yr	-0.0190774
spend_online_o_1yr	0.0178836
customer_type_2:treat_resid	-0.0163080
clicks_product_type_112_3yr	-0.0117976
orders_hm	0.0113834
spend_m_1yr:treat_resid	0.0113113
spend_d_k_1yr	0.0109901
spend_instore_p_1yr:treat_resid	0.0107071
clicks_product_type_001_12m	0.0098876
spend_instore_q_3yr:treat_resid	-0.0096798
mean_spend_attributed_mail_type_A:treat_resid	0.0087551
clicks_product_type_312_3yr	-0.0081961
clicks_product_type_201_3yr	0.0081121
spend_d_k_1yr:treat_resid	0.0080782
spend_h_1yr	0.0078652
web_activity_3m	0.0069211
spend_online_a_3yr	0.0066646
spend_d_h_1yr	0.0053205
spend_online_k:treat_resid	-0.0051437
spend_online_b_1yr	-0.0050362

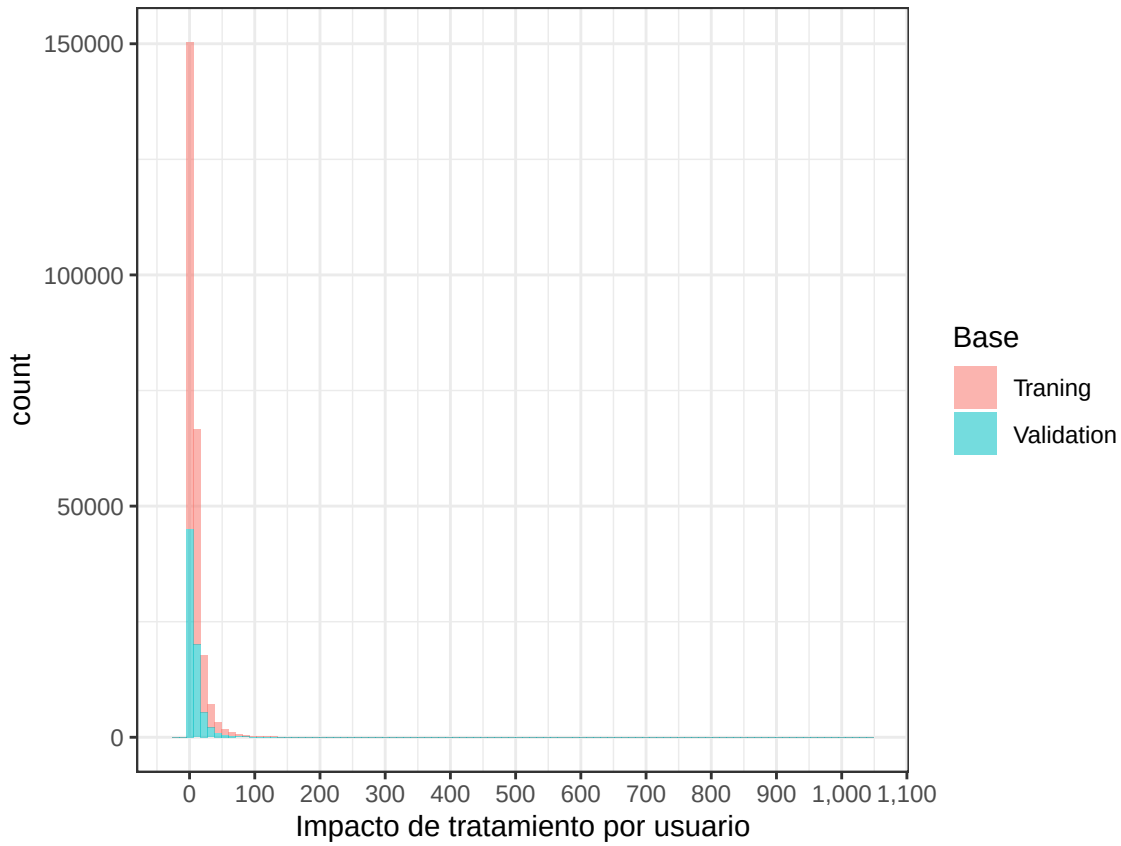
variable	coeficiente
spend_online_o_3yr	0.0049259
spent_q	0.0046833
emailreceived_months_since:treat_resid	0.0044344
spend_direct_1yr:treat_resid	0.0039164
spend_d_s_1yr:treat_resid	0.0038989
spend_h_3yr	0.0036893
spend_online_a_1yr	0.0032712
spend_online_sg_1yr	0.0032334
spend_online_sh	-0.0032142
spend_instore_o_3yr	0.0030384
spend_attributed_mail_type_B	0.0029430
spend_period_1b	0.0028114
spend_period_3b:treat_resid	0.0028104
spend_online_s:treat_resid	0.0028028
spend_online_b_1yr:treat_resid	-0.0027377
spend_period_2b	0.0027323
clicks_product_type_504	-0.0026925
spend_instore_t	0.0025407
spend_M_3yr:treat_resid	0.0025406
mean_spend_attributed_mail_type_B:treat_resid	0.0024125
spend_h_3yr:treat_resid	0.0021091
spend_instore_h_1yr:treat_resid	0.0018428
dollars_season_C	0.0016724
emailreceived_months_since	0.0014877
in_database_months	0.0014558
spend_l	0.0012996
mean_spend_attributed_mail_type_A	0.0010657
clicks_product_type_502_3m:treat_resid	0.0009926
spend_a_1yr:treat_resid	-0.0009390
orders_season_E:treat_resid	0.0008634
spend_e	-0.0008438
spend_m_1yr	0.0008404
spend_attributed_mail_type_C	0.0007883
spend_online_s_1yr:treat_resid	0.0007653
spend_instore_n:treat_resid	-0.0006874
emails_days_1yr:treat_resid	-0.0005932
spend_instore_o_3yr:treat_resid	-0.0005090
spend_z1	-0.0004938
spend_h:treat_resid	-0.0004849
spend_d_s_1yr	-0.0004519
spend_M_3yr	0.0004370
spend_z1:treat_resid	0.0004087
spend_direct_g	0.0004013
spend_online_a_1yr:treat_resid	-0.0002758
spend_notz_1yr	0.0002551
spend_online_g_1yr:treat_resid	-0.0002482
emailview_24m:treat_resid	-0.0002196
spend_instore_o	0.0002085
dollars_season_C:treat_resid	0.0001325
spend_h_1yr:treat_resid	0.0000227

10 (2 pts). Predice el CATE en la base de entrenamiento y en la base de validación. Como se ve la distribución

del impacto de tratamiento en ambas?

```
# Prediciendo en la base de entrenamiento
crm_training<-
  crm_training %>%
  mutate(ddml_impact = as.numeric(predict(ddml_hte, newdata = X , type = "response")))


# Validation
# Construyendo la primera X
X<-
  crm_validation %>%
  select(-c(customer_id, outcome_spend, training_sample, mailing_indicator))


# Generando treat_resid en validacion
scores_validation<-
  map_dfc(1:5,
                function(a) {
                   treat_hat<-as.numeric(predict(modelo[[a]]$treat_fit,
                                                 newdata = X,
                                                 type = "response"))

                   spend_hat<-as.numeric(predict(modelo[[a]]$spend_fit,
                                                 newdata = X,
                                                 type = "response"))

                   treat_resid <- crm_validation$mailing_indicator - treat_hat

                   spend_resid <- crm_validation$outcome_spend- spend_hat

                   scores<-bind_cols("treat_resid"= treat_resid, "spend_resid" = spend_resid)




                })


# Promediando los scores por K
scores_validation<-
  scores_validation %>%
  rowwise() %>%
  mutate(spend_resid = mean(spend_resid...2, spend_resid...4, spend_resid...6, spend_resid...8, spend_r
         treat_resid = mean(treat_resid...1, treat_resid...3, treat_resid...5, treat_resid...7, treat_r
  select(spend_resid, treat_resid)

# Treat resid en la base
crm_validation <-
  bind_cols(crm_validation, scores_validation %>% select(treat_resid))

X<-
  crm_validation %>%
  select(-c(customer_id, outcome_spend, training_sample, mailing_indicator))
```

```r
X<-sparse.model.matrix(~. + 0+ . *treat_resid, data = X)

crm_validation<-
  crm_validation %>%
  mutate(ddml_impact = as.numeric(predict(ddml_hte, newdata = X, type = "response")))


# Grafica de ambos
tau_i<-bind_rows(crm_training %>% select(customer_id, ddml_impact, training_sample, outcome_spend, maili
                 crm_validation %>% select(customer_id, ddml_impact, training_sample, outcome_spend, mai

tau_i<-
  tau_i %>%
  mutate(training_sample = if_else(training_sample ==1, "Traning", "Validation"))

ggplot(tau_i, aes(ddml_impact))+
  geom_histogram(bins = 100, aes(fill = training_sample), alpha = 0.55)+
  theme_bw()+
  labs(x = 'Impacto de tratamiento por usuario', fill = 'Base')+
  scale_x_continuous(labels = scales::comma, breaks = seq(0,1100,100))
```




```r
rm(X, coeficientes, coeficientes_g, coeficientes_m, mailing_ate, mailing_ate2)
```

####Causal Forest

11 (2pts). Ahora vayamos al causal forest. Estima un causal forest en la base de entrenamiento (Estima 750

árboles)

```
outcome_spend<-crm_training$outcome_spend

X<-
  crm_training %>%
  select(-c(customer_id, outcome_spend, training_sample, mailing_indicator, ddml_impact, treat_resid))

treat<-crm_training$mailing_indicator

a<-Sys.time()

causal_hte<-causal_forest(X = X, Y = outcome_spend, W = treat, num.trees = 750)

Sys.time() - a
```

```
Time difference of 16.22544 mins
```

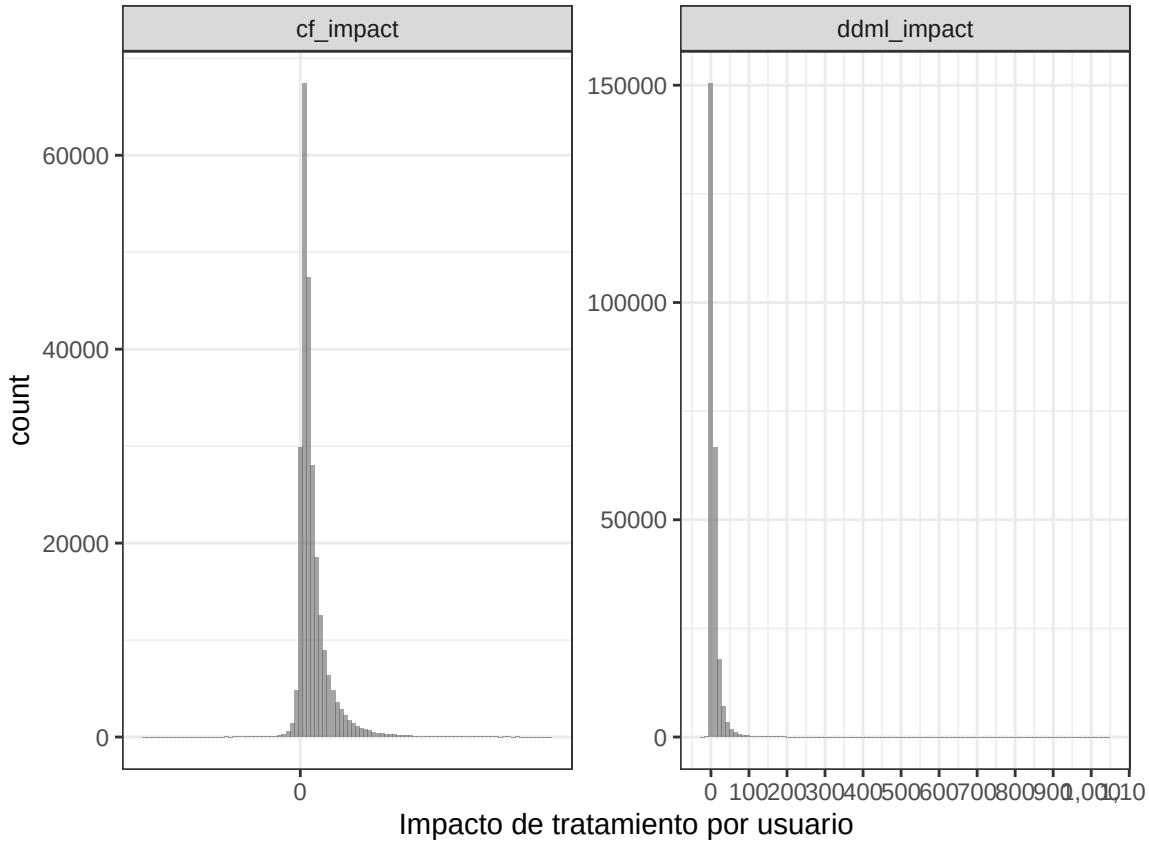
12 (3pts). Cómo se distribuye el impacto de tratamiento? Cuál es el impacto de tratamiento (ATE)? Qué tanto se acerca al impacto de tratamiento "real"? Cómo se compara con el impacto estimado con el ddml simple?

```
# ATE
average_treatment_effect(causal_hte)
```

```
 estimate   std.err
2.7077264 0.2073587
```

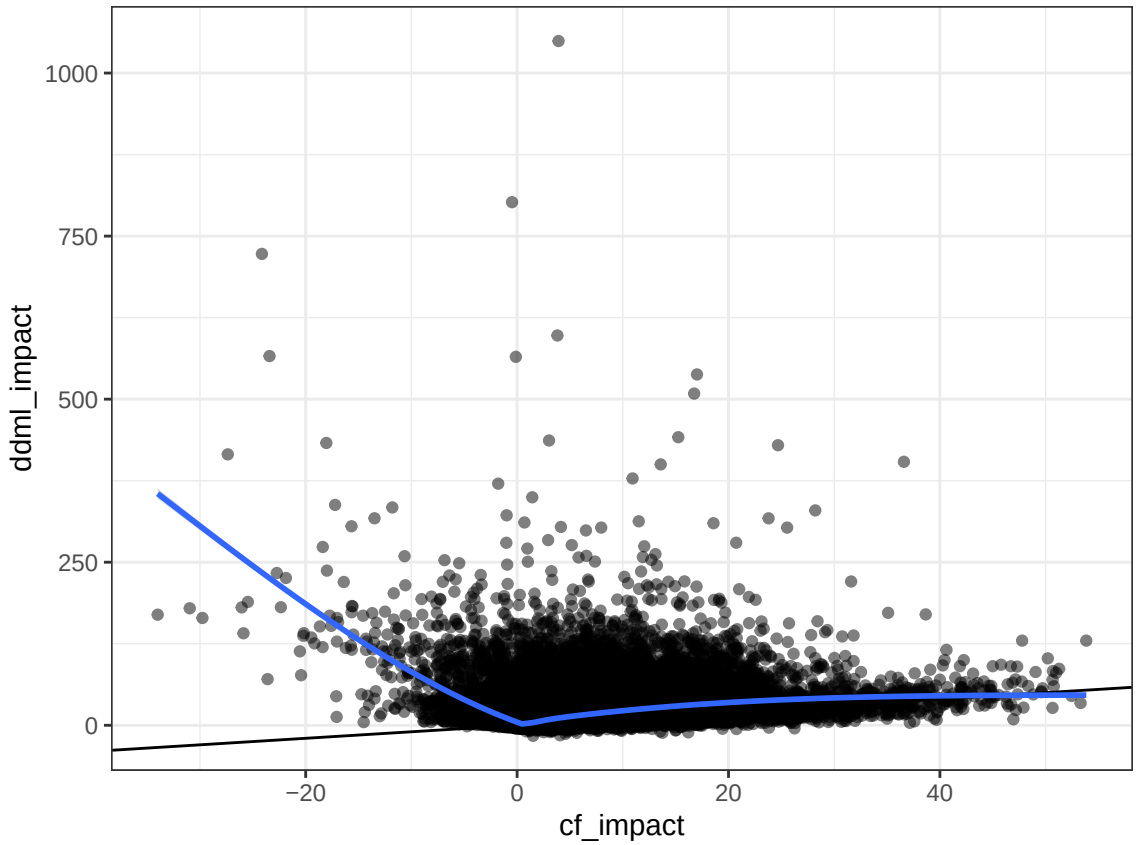
```
crm_training<-
  crm_training %>%
  mutate(cf_impact = predict(causal_hte)$predictions,
         spend_resid  = scores$spend_resid)


# Validation
X<-
  crm_validation %>%
  select(-c(customer_id, outcome_spend, training_sample, mailing_indicator, ddml_impact, treat_resid))

crm_validation<-
  crm_validation %>%
  mutate(cf_impact = predict(causal_hte, newdata = X)$predictions)


# Grafica de ambos
tau_i<-bind_rows(crm_training %>% select(customer_id, ddml_impact, training_sample, outcome_spend, maili
                 crm_validation %>% select(customer_id, ddml_impact, training_sample, outcome_spend, mai


tau_i_long<-
  tau_i %>%
  pivot_longer(cols = c(ddml_impact, cf_impact), names_to = "modelo", values_to = "estimador")

ggplot(tau_i_long, aes(estimador))+
  geom_histogram(bins = 100, aes(fill = training_sample), alpha = 0.55)+
  theme_bw()+
```

```
  labs(x = 'Impacto de tratamiento por usuario', fill = 'Base')+
  scale_x_continuous(labels = scales::comma, breaks = seq(0,1100,100))+
  facet_wrap(~modelo, scales = "free")
```



13. Haz un scatter plot de las predicciones de ambos modelos? Hay alguna relación?

```
ggplot(tau_i, aes(cf_impact, ddml_impact))+geom_point(alpha = 0.5)+
  geom_abline(slope = 1, intercept = 0)+theme_bw()+
  geom_smooth()
```




```
comparativa<-lm(ddml_impact ~ cf_impact, data = tau_i)

stargazer(comparativa)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Sun, May 09, 2021 - 22:37:49

Table 10:

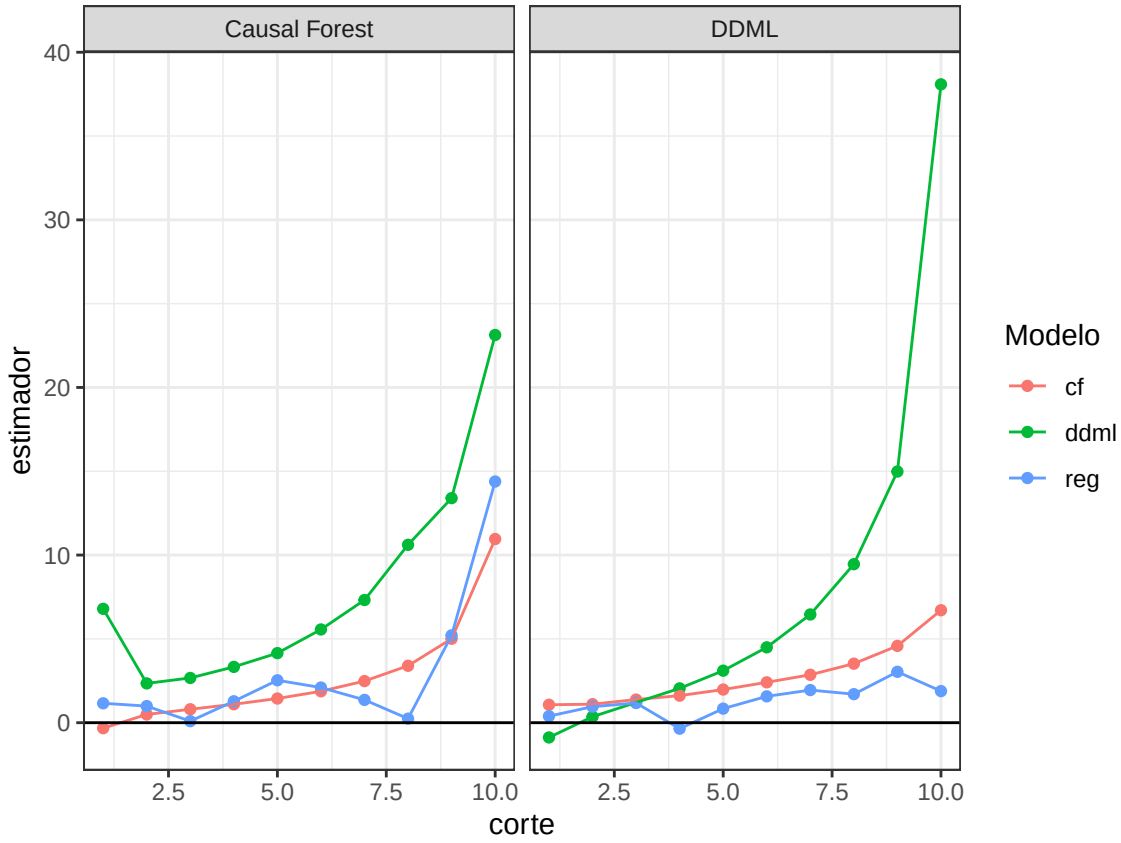
	Dependent variable:
	ddml_impact
cf_impact	1.480***
	(0.007)
Constant	3.944***
	(0.033)
Observations	250,000
R^2	0.140
Adjusted R^2	0.140
Residual Std. Error	13.344 (df = 249998)
F Statistic	40,623.010*** (df = 1; 249998)
Note:	*p<0.1; **p<0.05; ***p<0.01

14 (4pts). Evalúa el poder predictivo de cada modelo (OOS). Esto se hace por modelo: Divide la muestra en

10 partes con base en el score de ddml. Para cada parte, estima el impacto de tratamiento vía una regresión y saca el promedio del score. Valida si para los grupos que dice el score el impacto será más grande, el coeficiente de la regresión es. Cómo se ven los modelos? Cuál parece ser mejor?

```
tau_validation <-
  tau_i %>%
  filter(training_sample == 0) %>%
  mutate(cut_ddml = ntile(ddml_impact, 10),
         cut_cf = ntile(cf_impact, 10))


# DDML
por_ddml<-tau_validation %>% split(.$cut_ddml)

por_ddml<-
  map_dfr(por_ddml,
          function(x) {
            ddml_mean = mean(x$ddml_impact)
            cf_mean = mean(x$cf_impact)
            regresion = coef(lm(outcome_spend~mailing_indicator, data = x))["mailing_indicator"]

            tabla<-tibble(corte = first(x$cut_ddml), "ddml" = ddml_mean, "cf" = cf_mean, "reg" = regres

          })


# CAUSAL FOREST
por_cf<-tau_validation %>% split(.$cut_cf)

por_cf<-
  map_dfr(por_cf,
          function(x) {
            ddml_mean = mean(x$ddml_impact)
            cf_mean = mean(x$cf_impact)
            regresion = coef(lm(outcome_spend~mailing_indicator, data = x))["mailing_indicator"]

            tabla<-tibble(corte = first(x$cut_cf), "ddml" = ddml_mean, "cf" = cf_mean, "reg" = regresio

          })

comparativa<-bind_rows(por_ddml, por_cf, .id = "modelo")

comparativa<-comparativa %>% mutate(modelo = if_else(modelo==1, "DDML", "Causal Forest"))

comparativa<-
  comparativa %>%
  pivot_longer(cols = c(ddml, cf, reg),
               names_to = "Modelo", values_to = "estimador")

ggplot(comparativa, aes(corte, estimador, fill = Modelo, color = Modelo))+
  geom_point()+
  geom_line()+
  facet_wrap(~modelo)+
  theme_bw()+
```

```
  geom_hline(yintercept = 0)
```



15 (6 pts). Construye una estrategia de focalización a nivel usuario con base a los resultados de cada modelo. Considera lo siguiente:

- El costo marginal de mandar el mail es 0.99 USD
- El Beneficio marginal es el impacto incremental la utilidad generada por esas ventas
- El margen de ganancia sobre las ventas es de 32.5 fijo

Con esto, indica:

- Cuantos usuarios entrarían a la campaña?
- A partir de cuánto lift (ventas incrementales) entran?
- Cuál es el impacto promedio esperado de tu población final?
- Cuánta utilidad haremos con esta estrategia? Cómo se compara con la utilidad de la campaña sin focalizar?

```
# Utilidad marginal
tau_i<-
  tau_i %>%
  mutate(umg_ddml= ddml_impact*0.325-0.99,
         umg_cf = cf_impact*0.325-0.99)

# 71561
tau_i_cf<-
  tau_i %>%
```

```
  filter(umg_cf >=0)

# 138,234
tau_i_ddml<-
  tau_i %>%
  filter(umg_ddml>=0)

# Impacto minimo y promedio
str_c("Impacto mínimo DDML", round(min(tau_i_ddml$ddml_impact), digits = 2))
```

```
[1] "Impacto mínimo DDML3.05"
```

```
# Impacto minimo y promedio
str_c("Impacto mínimo CF", round(min(tau_i_cf$cf_impact), digits = 2))
```

```
[1] "Impacto mínimo CF3.05"
```

```
# Impacto promedio
str_c("Impacto promedio DDML", round(mean(tau_i_ddml$ddml_impact), digits = 2))
```

```
[1] "Impacto promedio DDML13.68"
```

```
# Impacto  promedio
str_c("Impacto promedio CF", round(mean(tau_i_cf$cf_impact), digits = 2))
```

```
[1] "Impacto promedio CF6.7"
```

```
# Impacto total
str_c("Impacto total DDML", scales::comma(round(sum(tau_i_ddml$ddml_impact), digits = 2)))
```

```
[1] "Impacto total DDML1,890,995"
```

```
# Impacto  total
str_c("Impacto total CF", scales::comma(round(sum(tau_i_cf$cf_impact), digits = 2)))
```

```
[1] "Impacto total CF477,958"
```

```
# Impacto total
str_c("Impacto total sin focalizar", scales::comma((average_treatment_effect(causal_hte)[1])*nrow(tau_i
```

```
[1] "Impacto total sin focalizar452,775"
```

```
# Utilidad total
str_c("Utilidad total DDML", scales::comma(round(sum(0.32*tau_i_ddml$ddml_impact)-nrow(tau_i_ddml)*0.99
```

```
[1] "Utilidad total DDML468,267"
```

```
# Utilidad total
str_c("Utilidad total CF", scales::comma(round(sum(0.32*tau_i_cf$cf_impact)-nrow(tau_i_cf)*0.99, digits
```

```
[1] "Utilidad total CF82,337"
```

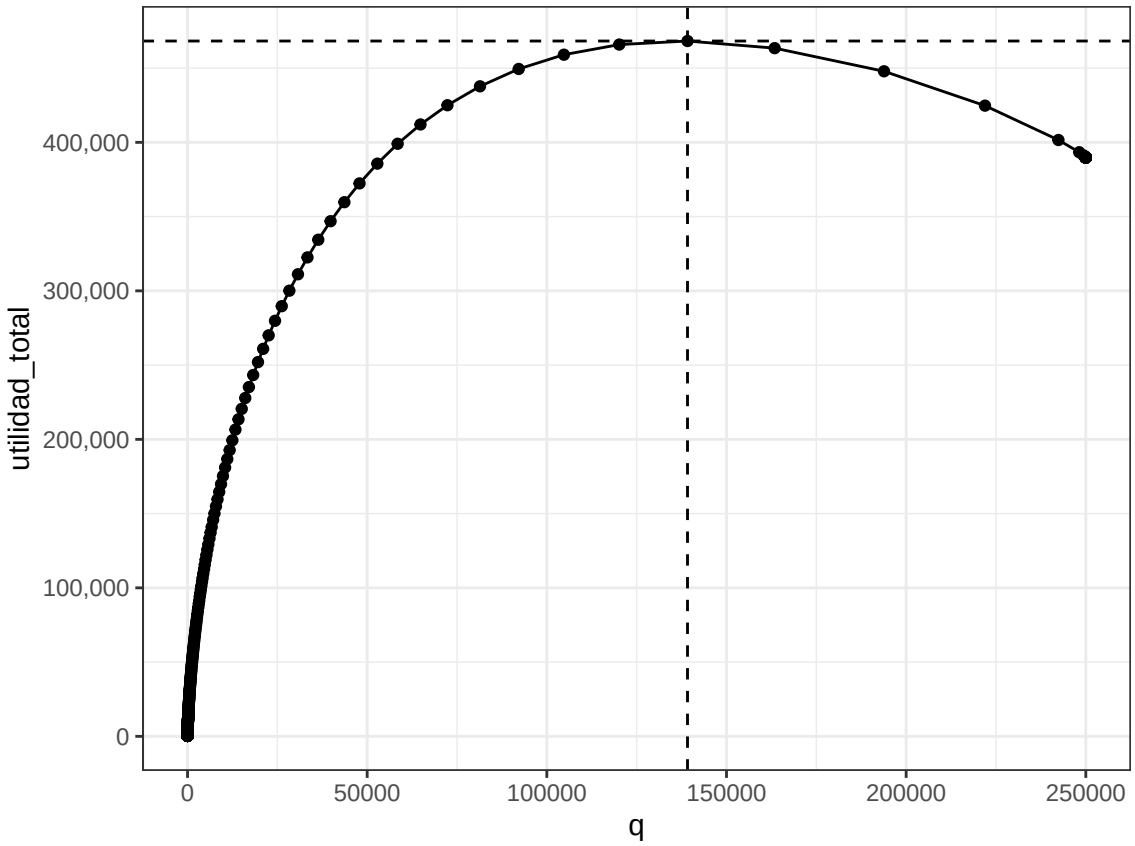
```
# Utilidad total sin focalizar
str_c("Utilidad total sin focalizar", scales::comma((0.32*average_treatment_effect(causal_hte)[1]-0.99)
```

```
[1] "Utilidad total sin focalizar-20,656"
```

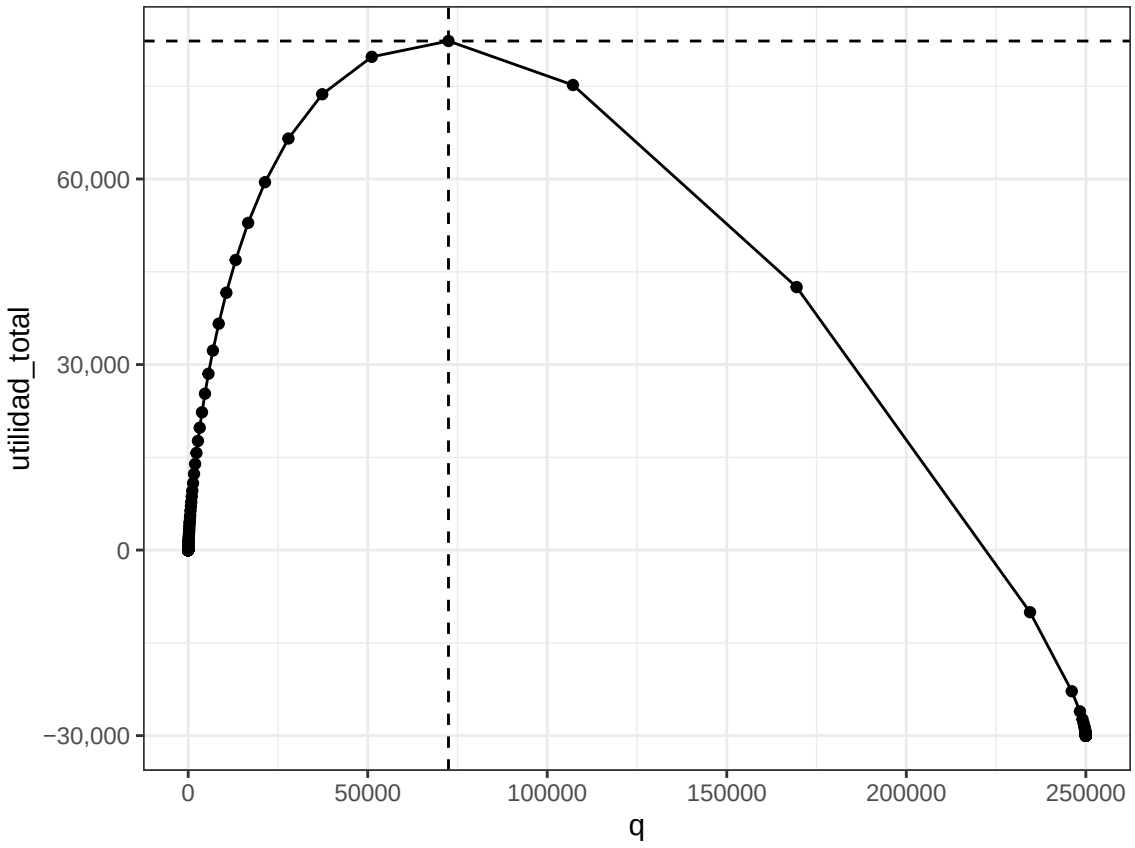
16 (3pts). Haz una gráfica del la utilidad total vs q (personas que entran en la campaña) para DDML y CF

```
cortes<-seq(-34,1000,1)
ut_q_ddml<-map_dfr(cortes,
                   function(x) {
```

```
                tau_i %>%
                  filter(ddml_impact>=x) %>%
                  summarise(beneficio_total = sum(ddml_impact),
                            q = n(),
                            costo_total = q*0.99,
                            utilidad_total = 0.32*beneficio_total - costo_total)

              })


ut_q_cf<-map_dfr(cortes,
              function(x) {
                tau_i %>%
                  filter(cf_impact>=x) %>%
                  summarise(beneficio_total = sum(cf_impact),
                            q = n(),
                            costo_total = q*0.99,
                            utilidad_total = 0.32*beneficio_total - costo_total)

              })



ggplot(ut_q_ddml, aes(q, utilidad_total))+geom_point()+
  geom_line()+
  theme_bw()+
  scale_y_continuous(labels = scales::comma)+
  geom_hline(yintercept = max(ut_q_ddml$utilidad_total), linetype= 'dashed')+
  geom_vline(xintercept = ut_q_ddml$q[ut_q_ddml$utilidad_total==max(ut_q_ddml$utilidad_total)], linetyp
```




```
ggplot(ut_q_cf, aes(q, utilidad_total))+geom_point()+
  geom_line()+
  theme_bw()+
  scale_y_continuous(labels = scales::comma)+
  geom_hline(yintercept = max(ut_q_cf$utilidad_total), linetype= 'dashed')+
  geom_vline(xintercept = ut_q_cf$q[ut_q_cf$utilidad_total==max(ut_q_cf$utilidad_total)], linetype= 'das
```


utilidad_total
60,000
30,000
0
−30,000
0
50000
100000
150000
200000
250000
q