Trabajo Práctico 1

Alcantara Fuentes, Victor Alan Bocco, Alessio
Buscaglia, Florencia Paula Ojeda, Rodrigo Nicolás

30 April, 2022

Contents

1 Introducción

El presente informe corresponde al Trabajo Práctico 1 de Métodos Estadísticos Aplicados a los Negocios. El objetivo del mismo es la evaluación del efecto de la comunicación entre compradores y vendedores en la plataforma eBay sobre las probabilidades de venta. El informe se estructura en tres secciones principales y un anexo. La primera consiste en un análisis exploratorio de los datos para la caracterización de las variables y la identificación de *outliers*. Luego, el profiling finaliza con un análisis univariado y bivariado de las variables más pertinentes para el objetivo del estudio. La segunda sección contiene un análisis empírico del efecto de la introducción de nuevas metodologías de comunicación entre oferentes y demandantes. Por último, el reporte concluye con la aplicación de un modelo estadístico

para la estimación del efecto de estas nuevas metodologías sobre las ventas. En el anexo se incluyen gráficos accesorios y una copia del código utilizado para la obtención de los resultados mostrados.

2 Exploración del dataset

2.1 Estructura del dataset

El dataset contiene 13 variables cuya descripción se detalla a continuación.

- date: fecha
- itemid: id del producto
- buyerid: id del comprador
- itemsold: =1 si se vendió el producto, =0 en caso contrario
- message: =1 si el comprador envió un mensaje, =0 en caso contrario
- desktop: =1 si el comprador usó la versión desktop (A), =0 si usó la versión móvil (B)
- post: =1 a partir del 23 de mayo de 2016, =0 antes de dicha fecha
- category: categoría del producto en venta
- condition: =1 si es nuevo, =0 si es usado
- askingprice: precio ofrecido
- holiday: =1 si es feriado, =0 en caso contrario
- temp: temperatura
- precipitation: precipitaciones

2.1.1 Tipos de variables

La Tabla 1 muestra los tipos de variables presentes en el dataset. Para cada una de ellas se muestra el tipo de dato y la cantidad de faltantes y valores únicos.

Table 1: Tipos de variables presentes.

Variables	Tipo	Faltantes (N)	Faltantes (%)	Único (N)	Único (tasa)
date	Date	0	0	56	0.000280
itemid	numeric	0	0	5	0.000025
buyerid	numeric	0	0	493	0.002465
itemsold	numeric	0	0	2	0.000010
message	numeric	0	0	2	0.000010
desktop	numeric	0	0	2	0.000010
post	numeric	0	0	2	0.000010
category	character	0	0	5	0.000025
condition	numeric	0	0	2	0.000010
askingprice	numeric	0	0	36798	0.183990
holiday	numeric	0	0	2	0.000010
temp	numeric	0	0	25	0.000125
precipitation	numeric	0	0	14	0.000070

Del análisis exploratorio realizado, se observaron las siguientes situaciones:

- Tipos de variables:
  - Fecha: Date
  - Categórica: cateogry
  - Cuantitativas:
    - Continuas: itemid, buyerid, askingprice, temp, precipitation
    - Discretas: itemsold, message, desktop, post, condition, holiday
- Ninguna variable cuenta con valores faltantes.
- Si bien en la Tabla 1 algunas de las variables cuantitativas discretas se muestran como numéricas, en realidad se trata de variables dicotómicas que expresan la ausencia o presencia de determinada cuestión.

2.1.2 Variables cuantitativas

Dentro de las variables cuantitativas, la Tabla 2 muestra las principales medidas de resumen que caracterizan a cada una de ellas.

Table 2: Diagnóstico variables cuantitativas

variables	min	Q1	mean	median	Q3	max	zero	minus
itemid	1200000000.00	3400000000.00	5417668500.000000	5600000000.00	7800000000.00	9100000000.00	0	0
buyerid	7384.00	250000000.00	500117278.912840	500000000.00	750000000.00	1000000000.00	0	0
itemsold	0.00	0.00	0.479980	0.00	1.00	1.00	104004	0
message	0.00	0.00	0.070305	0.00	0.00	1.00	185939	0
desktop	0.00	0.00	0.551970	1.00	1.00	1.00	89606	0
post	0.00	0.00	0.500105	1.00	1.00	1.00	99979	0
condition	0.00	0.00	0.425750	0.00	1.00	1.00	114850	0
askingprice	2.53	5.06	92.725363	16.24	66.09	5874.36	0	0
holiday	0.00	0.00	0.090485	0.00	0.00	1.00	181903	0
temp	5.50	12.00	14.811587	16.50	18.00	22.50	0	0
precipitation	0.00	0.00	3.224925	0.00	5.00	45.00	110833	0

A partir de la Tabla se observa lo siguiente:

- itemdid y buyerid: corresponden a indentificadores unívocos de uso interno y carecen de valor para el presente estudio.
- Se confirma el carácter dictómico de las variables identificadas en el apartado anterior.
- La variable askingprice exhibe un fuerte sesgo a izquierda dado que la media es muy superior a la mediana. También tiene una importante disperión dada la diferencia entre el limite superior del intervalo intercuartílico y el máximo. No presenta valores negativos ni nulos por lo que es consistente con su significado. * Las variables meteorológicas *temp* y *precipitation* muestran un comportamiento consistente con lo que se esperaría para variables de este tipo.

2.1.3 Variables categóricas

La Tabla 3 muestra una tabla de contigencia con las principales métricas que describen cada categoría de la variable *category*

Table 3: Diagnóstico variables categóricas

Variable	Clases	N	Freq	Ratio	Ranking
category	Clothing, Shoes, & Accessories	200000	40206	20.1030	1
category	Toys & Hobbies	200000	40127	20.0635	2
category	Jewelry & Watches	200000	40029	20.0145	3
category	Books	200000	39841	19.9205	4
category	Computer/Tablets & Networking	200000	39797	19.8985	5

Las cinco clases tienen una cantidad de observaciones muy similar, con una frecuencia en torno al 20%.

2.1.4 Muestra del dataset

A continuación de muestra la Tabla 4 dónde se observan las 5 primeras filas del dataset.

Table 4: Muestra del dataset

date	itemid	buyerid	itemsold	message	desktop	post	category	condition	askingprice	holiday	temp	precipitation
2016-04-25	9100000000	200000000	0	0	1	0	Computer/Tablets & Networking	1	4.93	0	5.5	11
2016-04-25	9100000000	360000000	0	0	1	0	Computer/Tablets & Networking	1	15.23	0	5.5	11
2016-04-25	7800000000	81000000	1	0	1	0	Clothing, Shoes, & Accessories	1	33.81	0	5.5	11
2016-04-25	5600000000	160000000	0	0	1	0	Jewelry & Watches	0	51.79	0	5.5	11
2016-04-25	7800000000	27000000	0	0	1	0	Clothing, Shoes, & Accessories	0	10.07	0	5.5	11
2016-04-25	5600000000	930000000	0	0	1	0	Jewelry & Watches	1	30.91	0	5.5	11

A partir de éste primer análisis exploratorio y en función del objetivo del estudio se seleccionaron las variables *itemsold*, *desktop*, *post*, *condition*, *holiday*, *temp* y *precipitation*. Sobre las mencionadas variables se mostrará a continuación un análisis uni y bivariado y se evaluará la presencia de outliers en las variables cuantitativas seleccionadas.

2.2 Análisis univariado y bivariado

2.2.1 Cantidad de ventas por plataforma

La Figura 1 muestra la cantidad de ventas por tipo de plataforma. En el panel izquierdo se muestran la cantidad de operaciones totales y en el derecha la cantidad de ventas concretadas. Se observa una distribución bastante pareja entre ambas, con un 45% de las ventas en la versión móvil y un 55% en la versión de escritorio.

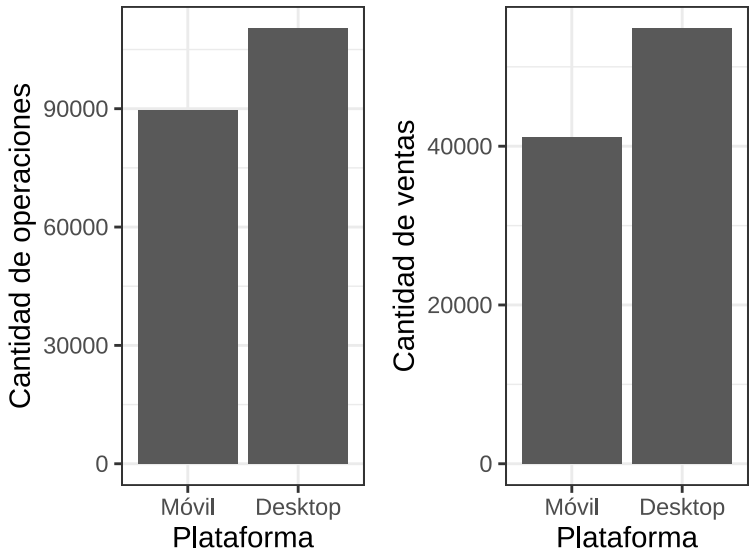

Figure 1: Ventas por tipo de plataforma.

2.2.2 Cantidad de ventas por momento de compra

La Figura 2 muestra la cantidad de operaciones antes y después del 23 de mayo de 2016, momento en que se permitió interacción entre compradores y vendedores. En el panel izquierdo se muestran la cantidad de operaciones totales y en el derecha la cantidad de ventas concretadas. se observa que la cantidad de operaciones total fue la misma para ambos momentos pero hubo un ligero aumento en las ventas concretadas a partir de la posibilidad de interacción.

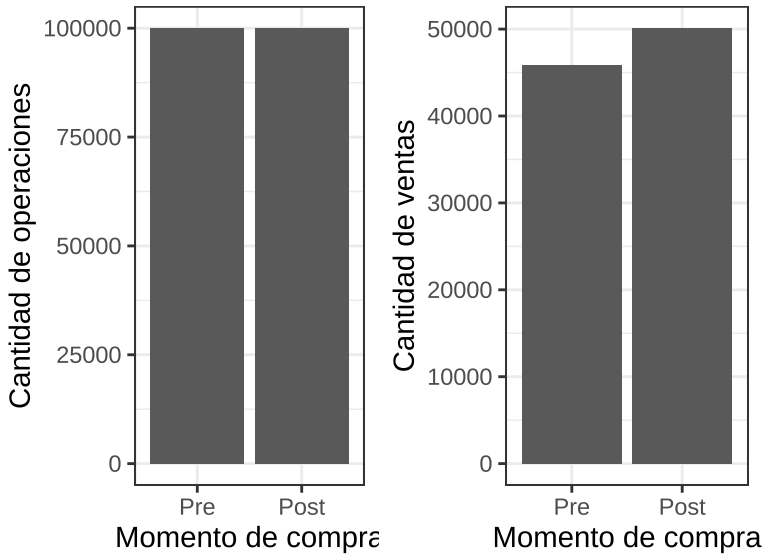

Figure 2: Ventas anteriores y posteriores al cambio en la plataforma.

2.2.3 Cantidad de ventas por condición del producto

La Figura 3 muestra la cantidad de operaciones (izquierda) y ventas (derecha) según la condición del producto, es decir, si se trata de un producto nuevo o usado.

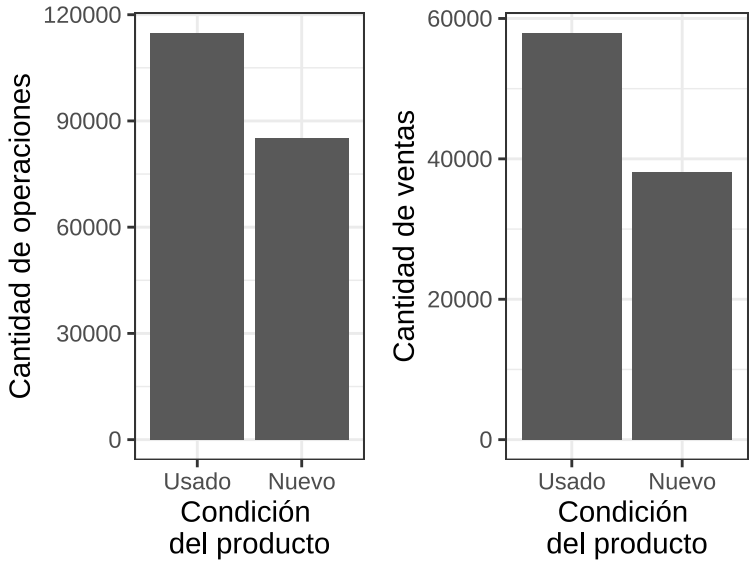

Figure 3: Ventas según la condición del producto.

2.2.4 Efecto de las condiciones climáticas

La Figura 4 muestra la cantidad de ventas y operaciones por percentil de temperatura. Es decir, se dividió la variable *temp* en 5 percentiles iguales y se contaron la cantidad de operaciones (izquierda) y ventas (derecha) que se llevaron a cabo en cada uno de ellos.

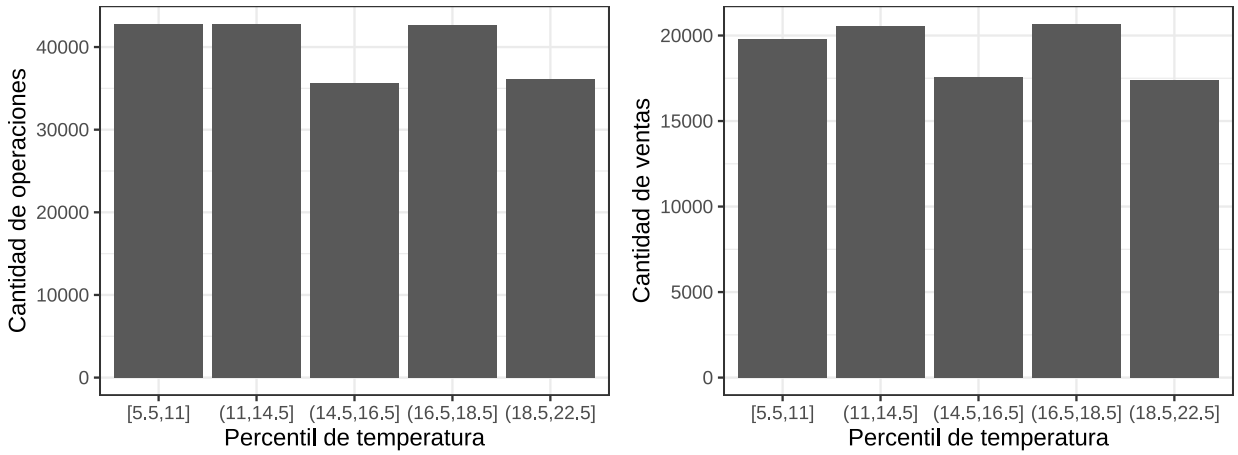

Figure 4: Ventas según el percentil de temperatura.

En la Figura no se observan diferencias significativas entre clases tanto para operaciones como para ventas concretadas.

2.2.5 Precipitación

La Figura 5 muestra la cantidad de ventas y operaciones por tipo de día. La Organización Meteorológica Mundial (OMM) clasifica a los días en lluviosos o secos si la precipitación acumulada diaria es superior a 0.5 mm. Si bien no se conoce la unidad de medida de la variable se asume que está expresada en el sistema métrico. Luego, se contaron la cantidad de operaciones (izquierda) y ventas (derecha) que se llevaron a cabo en cada uno de ellos.

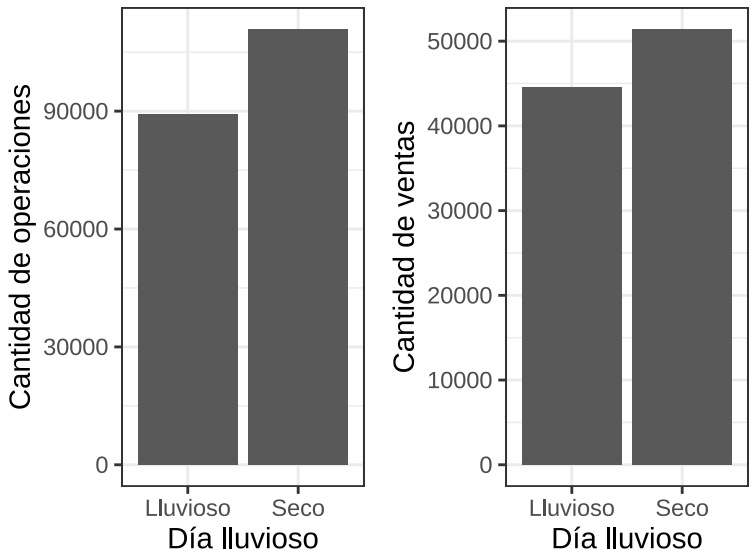

Figure 5: Ventas según el tipo de día.

En la Figura no se observan diferencias significativas entre clases tanto para operaciones como para ventas concretadas.

2.3 Análisis de outliers

Los outliers se identificaron para las variables cuantitativas continuas. La fórmula utilizada para fue la por defecto en R para el diagrama de cajas. La misma se detalla a continuación.

```
# Outliers inferiores
max(min(x), Q1 - (IQR(x)*1.5))
# Outliers superiores
min(max(x), Q3 + (IQR(x)*1.5))
```

La Tabla @ref(tab:out_tabla) muestra los resultados del análisis. Sólo se detectaron valores anómalos en *askingprice* pero dado que nor formará parte del análisis no se hizo más hincapié en ellos.

variables	outliers_cnt	outliers_ratio	outliers_mean	with_mean	without_mean
askingprice	28048	14.024	496.73982	92.725363	26.824429
temp	0	0.000	NaN	14.811587	14.811587
precipitation	10748	5.374	28.84518	3.224925	1.769899

Con respecto a la precipitación, esta variable tiene un comportamiento especial cuando se trata de valores diarios. Existen una gran cantidad de ceros por lo que es muy frecuente la identificación de outliers cuando en realidad no lo son. El análisis de valores extremos de lluvia está fuera del alcance del presente pero se muestra en la Figura 6 la distribución de la variable considerando los valores anómalos en el panel superior y luego de su eliminación en el inferior.

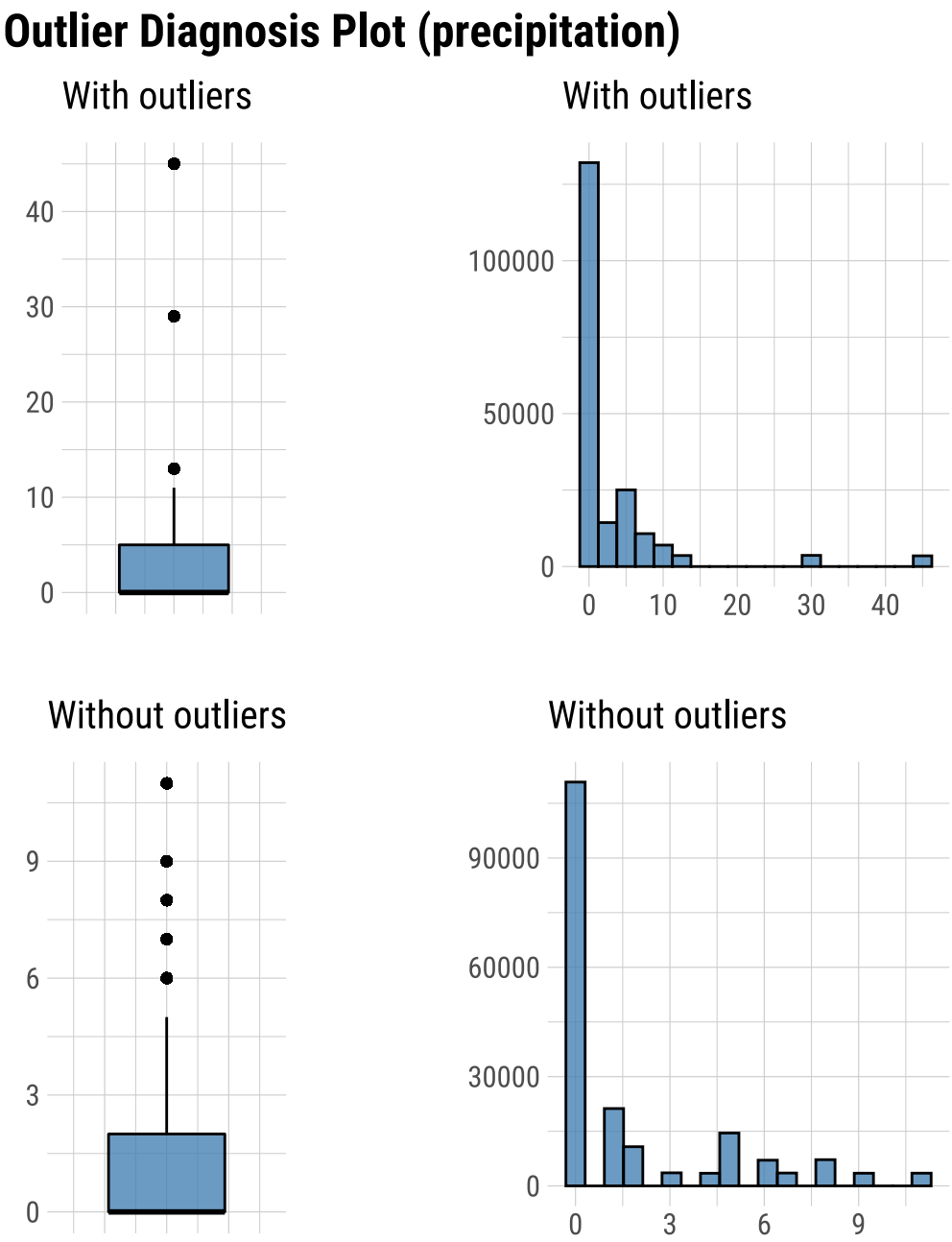

Figure 6: Outliers de la variable precipitación

3 Análisis de ventas

En la presente sección comienza el análisis del impacto en la ventas de la nueva estrategia de comunicación. En la 7 muestra la proporción de ventas para las dos plataformas (Escritorio y Móvil) y antes y después de permitir la interacción. El punto de la Figura corresponde a la proporción de ventas en cada combinación de plataforma/momento y las barras al intervalo

de confianza del 95% (percentiles).

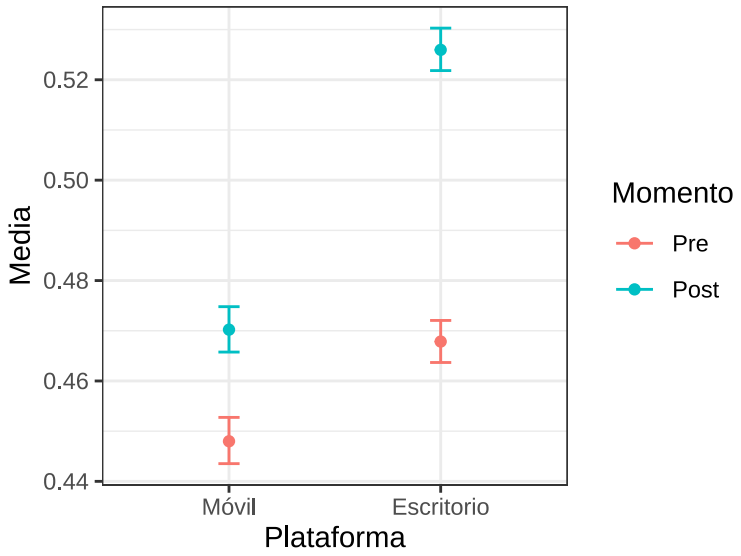

Figure 7: Proporción de las ventas por plataforma y momento de observación.

Del gráfico obtenido para los intervalos de confianza se observa la existencia de solapamiento en particular entre la proporción de ventas de la versión Móvil (después del 23.05) y la proporción de ventas de la versión Desktop (antes del 23.05). Tal situación podría generar ruido en la base de datos y no permitir que se pueda diferenciar el real impacto de la variable 'post' en el incremento de la probabilidad de que se realice una venta cuando el comprador y vendedor se comuniquen.

Luego de conocer las proporciones con sus respectivos intervalos de confianza se realizó una prueba de proporciones para conocer si el cambio impulsó la cantidad de ventas en la versión Desktop. La 5 muestra los resultados de dicha prueba.

Table 5: Resúmen de la prueba de hipótesis.

Media estimada	Estadístico	p.valor	método	Hipótesis alternativa
0.4878028	13.66092	0.9998905	1-sample proportions test without continuity correction	greater

De la prueba de hipótesis realizada se verificó que al 95% de confianza no hay evidencia para rechazar la hipótesis nula de que la proporción de ventas de productos nuevos en la versión Desktop después del 23 de mayo de 2016 no sea superior al 50%, por lo que se puede afirmar que tal proporción no es superior al 50%.

3.0.1 Potencia de la prueba

La 8 muestra la potencia de la prueba anterior para distintos tamaños de muestra utilizando un valor de α de 0.05.

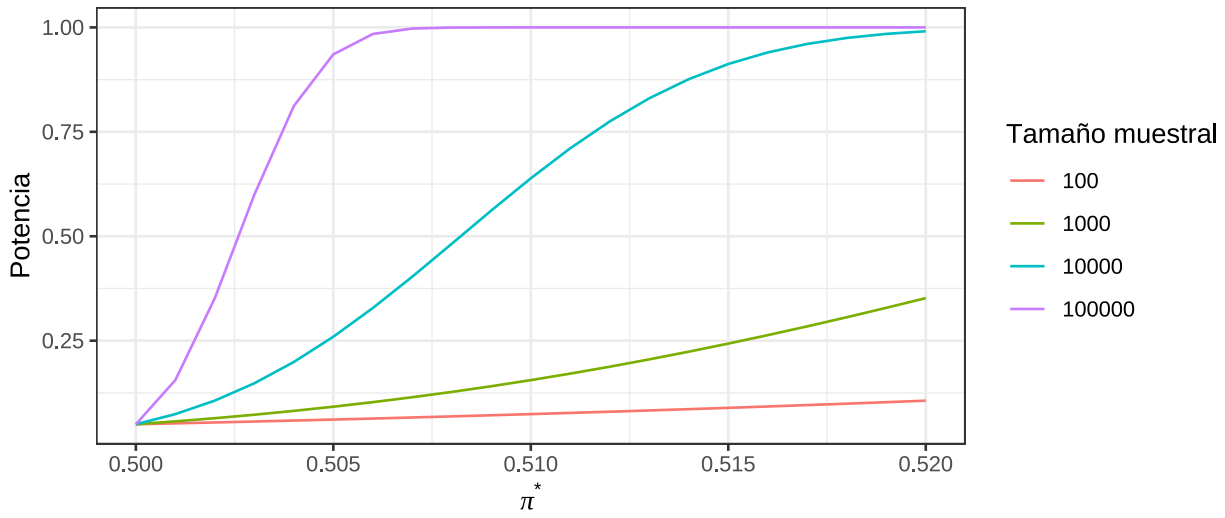

Figure 8: Potencia de la prueba de hipótesis para distintos tamaños muestrales.

Se observa el rápido crecimiento de la curva al aumentar el tamaño muestral en sintonía con la que se esperaría en base a la teoría.

3.1 Modelos de regresión

En la presente sección se muestran distintos modelos para explicar el efecto de distintas variables sobre la probabilidad de que se concrete una compra. Para ello se utilizará un modelo de probabilidad lineal (LPM, por sus siglas en inglés). Estos modelos son especialmente interesantes para explorar los efectos marginales sobre la variable de interés. A continuación se presentan distintos modelos con un grado creciente de complejidad.

3.1.1 Modelo básico

La ecuación del modelo básico se muestra en la ecuación 1. Las principales variables del modelo son *desktop* y *post*.

$$\text{itemsold} = \beta_0 + \beta_1(\text{desktop}) + \beta_2(\text{post}) + \beta_3(\text{desktop} \times \text{post}) + \epsilon \tag{1}$$

Los resultados del modelo de muestran en la tabla siguiente.

	Model 1
(Intercept)	0.448
	[0.443, 0.453]
	s.e. = 0.002
	t = 190.730
	p = 0.000
desktop	0.020
	[0.014, 0.026]
	s.e. = 0.003
	t = 6.272
	p = 0.000
post	0.022
	[0.016, 0.029]
	s.e. = 0.003
	t = 6.679
	p = 0.000
desktop × post	0.036
	[0.027, 0.045]
	s.e. = 0.004
	t = 8.000
	p = 0.000
Num.Obs.	200 000
R2	0.004
R2 Adj.	0.003
AIC	289 303.1
BIC	289 354.2
Log.Lik.	−144 646.570
F	234.527
Std.Errors	Robust

Mientras que el modelo ajustado se muestra en la ecuación 2.

$$\hat{\text{items}}\text{old} = 0.448 + 0.02(\text{desktop}) + 0.022(\text{post}) + 0.036(\text{desktop} \times \text{post}) \tag{2}$$

De la estimación realizada, se observa que, manteniendo lo demás constante, el hecho de utilizar la versión Desktop de la plataforma incrementa la probabilidad de que se realice una venta en 1.98 puntos porcentuales (β_1); el hecho de que se esté en un momento a partir del 23.05.2016, incrementa la probabilidad de que se realice una venta en 2.22 puntos porcentuales (β_2); y el hecho de que se utilice la versión Desktop de la plataforma y a la vez se esté en un momento a partir del 23.05.2016, incrementa la probabilidad de que se realice una venta en 3.58 puntos porcentuales β_3). Por otro lado, el hecho que el comprador y vendedor tengan la probabilidad de comunicarse incrementa la probabilidad de que se concrete una venta en 7.79 puntos porcentuales ($\beta_1 + \beta_2 + \beta_3$).

Además del análisis del modelo básico original se realizaron 1000 replicaciones del mismo para evaluar la estabilidad de los coeficientes. Los resultados del bootstrap se muestra en la Figura 9.

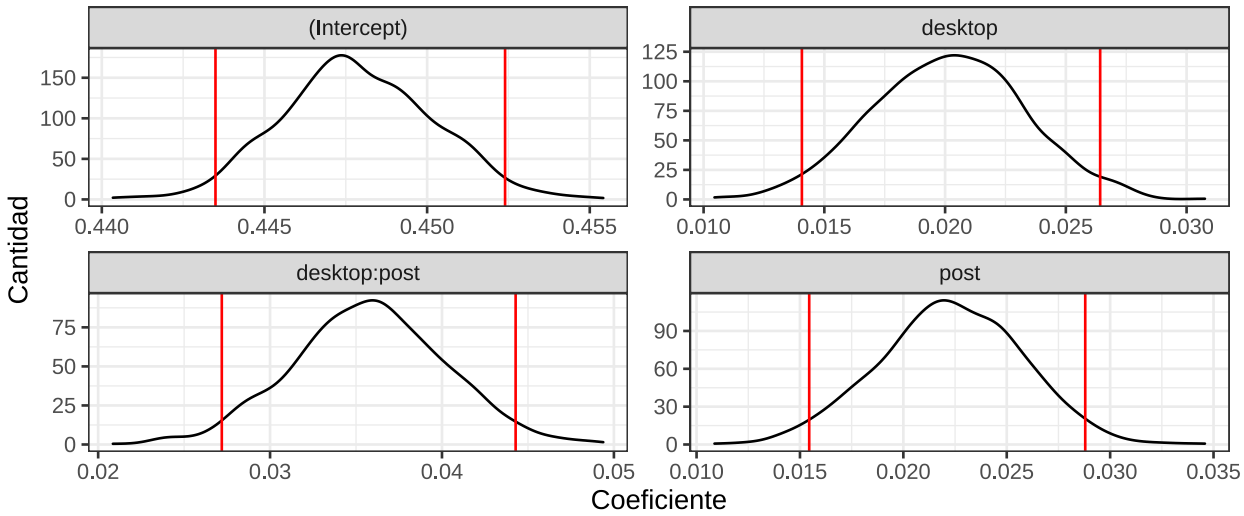

Figure 9: Coeficientes a partir del remuestreo del modelo de regresión básico.

En la Figura se muestra la densidad de los coeficientes en negro mientras que las barras rojas corresponden al intervalo de confianza del 95% calculado a partir de los percentiles.

3.1.2 Modelo básico + condición del producto

La ecuación del modelo básico se muestra en la ecuación 1. Las principales variables del modelo son *desktop* y *post*. La ecuación del modelo se muestra en la ecuación 3. A diferencia del caso anterior, se crean dos dataset filtrando por condición del producto. Es decir, se corre el modelo sobre los productos nuevos y otro modelo sobre los productos usados.

$$\text{itemsold} = \beta_0 + \beta_1(\text{desktop}) + \beta_2(\text{post}) + \beta_3(\text{desktop} \times \text{post}) + \epsilon \tag{3}$$

Los resultados del modelo de muestran en la tabla siguiente.

	Modelo nuevo	Modelo usado
(Intercept)	0.425	0.466
	[0.418, 0.432]	[0.460, 0.472]
	s.e. = 0.004	s.e. = 0.003
	t = 120.230	t = 148.419
	p = 0.000	p = 0.000
desktop	0.005	0.029
	[−0.005, 0.014]	[0.021, 0.038]
	s.e. = 0.005	s.e. = 0.004
	t = 1.017	t = 7.021
	p = 0.309	p = 0.000
post	0.016	0.027
	[0.006, 0.026]	[0.019, 0.036]
	s.e. = 0.005	s.e. = 0.004
	t = 3.130	t = 6.138
	p = 0.002	p = 0.000
desktop × post	0.042	0.031
	[0.029, 0.055]	[0.019, 0.042]
	s.e. = 0.007	s.e. = 0.006
	t = 6.167	t = 5.164
	p = 0.000	p = 0.000
Num.Obs.	85 150	114 850
R2	0.003	0.004
R2 Adj.	0.003	0.004
AIC	122 425.1	166 232.2
BIC	122 471.9	166 280.4
Log.Lik.	−61 207.553	−83 111.081
Std.Errors	Robust	Robust

La ecuación del modelo ajustado con datos de productos nuevos es:

$$\hat{\text{items}}\text{old} = 0.43 + 0(\text{desktop}) + 0.02(\text{post}) + 0.04(\text{desktop} \times \text{post}) \tag{4}$$

La ecuación del modelo ajustado con datos de productos usados es:

$$\hat{\text{items}}\text{old} = 0.47 + 0.03(\text{desktop}) + 0.03(\text{post}) + 0.03(\text{desktop} \times \text{post}) \tag{5}$$

Para productos Nuevos:

De la estimación realizada, se observa que, manteniendo lo demás constante, para productos nuevos el hecho de utilizar la versión Desktop de la plataforma incrementa la probabilidad de que se realice una venta en 0.48 puntos porcentuales (β_1); el hecho de que se esté en un momento a partir del 23.05.2016, incrementa la probabilidad de que se realice una venta en

1.56 puntos porcentuales (β_2); y el hecho de que se utilice la versión Desktop de la plataforma y a la vez se esté en un momento a partir del 23.05.2016, incrementa la probabilidad de que se realice una venta en 4.21 puntos porcentuales Beta 3). Por otro lado, el hecho que el comprador y vendedor tengan la probabilidad de comunicarse incrementa la probabilidad de que se concrete una venta en 6.26 puntos porcentuales ($\beta_1 + \beta_2 + \beta_3$). Comparando con punto anterior (pregunta N° 5), se observa que la estimación, considerando productos nuevos, genera un incremento de probabilidad menor (6.26 vs 7.79) de que se concrete una venta por el hecho de que el comprador y vendedor tengan la probabilidad de comunicarse. Asimismo, en este caso se observa que la variable 'desktop' no es estadísticamente significativa.

Para productos Usados:

De la estimación realizada, se observa que, manteniendo lo demás constante, para productos usados el hecho de utilizar la versión Desktop de la plataforma incrementa la probabilidad de que se realice una venta en 2.94 puntos porcentuales (β_1); el hecho de que se esté en un momento a partir del 23.05.2016, incrementa la probabilidad de que se realice una venta en 2.72 puntos porcentuales (β_2); y el hecho de que se utilice la versión Desktop de la plataforma y a la vez se esté en un momento a partir del 23.05.2016, incrementa la probabilidad de que se realice una venta en 3.06 puntos porcentuales Beta 3). Por otro lado, el hecho que el comprador y vendedor tengan la probabilidad de comunicarse incrementa la probabilidad de que se concrete una venta en 8.7 puntos porcentuales ($\beta_1 + \beta_2 + \beta_3$). Comparando con punto anterior (pregunta N° 5), se observa que la estimación, considerando productos usados, genera un incremento de probabilidad mayor (8.7 vs 7.79) de que se concrete una venta por el hecho de que el comprador y vendedor tengan la probabilidad de comunicarse.

Al igual que en el caso anterior se realizó un remuestreo de ambos modelos para conocer la estabilidad de los coeficientes. Los resultados del bootstrap se muestra en la Figura 10.

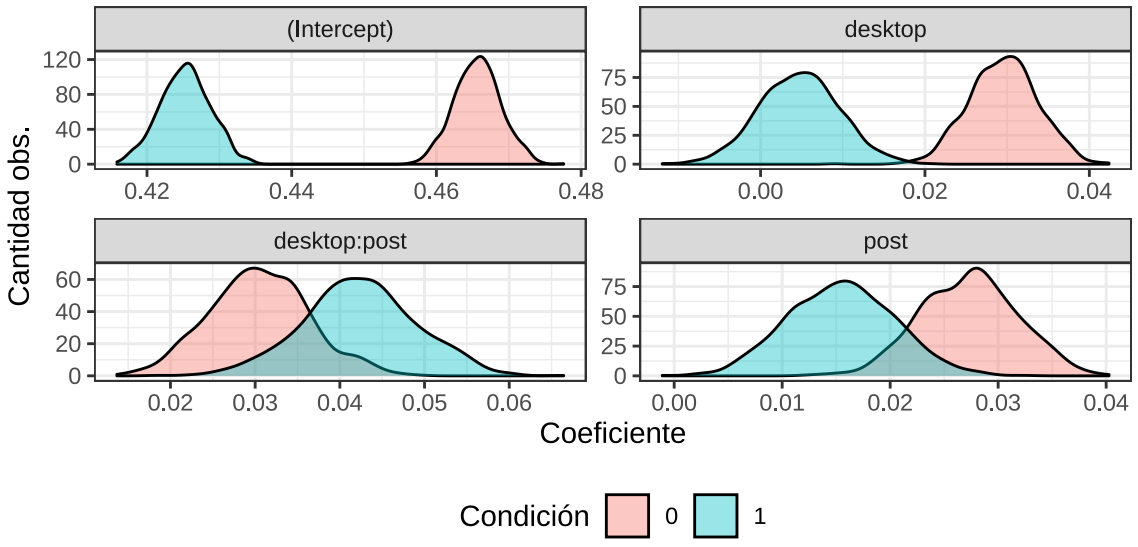

Figure 10: Coeficientes a partir del remuestreo del modelo de regresión discriminando por condición

3.1.3 Modelo básico + condiciones meteorológicas

La ecuación del modelo es similar al modelo básico pero se incorporan las variables meteorológicas de temperatura y precipitación. Se ha realizado dos tipos de comparaciones. Por un lado se evalúo el modelo climático vs el básico y en una segunda instancia se comparó discriminando la condición del producto mostrada en el apartado anterior. La ecuación del modelo climático se muestra a continuación.

$$\text{itemsold} = \beta_0 + \beta_1(\text{desktop}) + \beta_2(\text{post}) + \beta_3(\text{precipitation}) + \beta_4(\text{temp}) + \beta_5(\text{desktop} \times \text{post}) + \epsilon \tag{6}$$

Los resultados del modelo de muestran en la tabla siguiente.

	Modelo básico	Modelo clima
(Intercept)	0.448	0.450
	[0.443, 0.453]	[0.442, 0.459]
	s.e. = 0.002	s.e. = 0.004
	t = 190.730	t = 100.601
	p = 0.000	p = 0.000
desktop	0.020	0.020
	[0.014, 0.026]	[0.014, 0.026]
	s.e. = 0.003	s.e. = 0.003
	t = 6.272	t = 6.272
	p = 0.000	p = 0.000
post	0.022	0.015
	[0.016, 0.029]	[0.008, 0.022]
	s.e. = 0.003	s.e. = 0.004
	t = 6.679	t = 4.193
	p = 0.000	p = 0.000
desktop × post	0.036	0.036
	[0.027, 0.045]	[0.027, 0.045]
	s.e. = 0.004	s.e. = 0.004
	t = 8.000	t = 7.978
	p = 0.000	p = 0.000
precipitation		0.002
		[0.002, 0.002]
		s.e. = 0.000
		t = 12.414
		p = 0.000
temp		0.000
		[−0.001, 0.000]
		s.e. = 0.000
		t = −1.171
		p = 0.241
Num.Obs.	200 000	200 000
R2	0.004	0.004
R2 Adj.	0.003	0.004
AIC	289 303.1	289 152.9
BIC	289 354.2	289 224.4
Log.Lik.	−144 646.570	−144 569.462
F	234.527	172.263
Std.Errors	Robust	Robust

Las ecuación del modelo es la siguiente:

$$\hat{\text{item}}\text{sold} = 0.45 + 0.02(\text{desktop}) + 0.02(\text{post}) + 0(\text{precipitation}) + 0(\text{temp}) + 0.04(\text{desktop} \times \text{post}) \tag{7}$$

Las estimaciones de los coeficientes cambian sobre todo en el coeficiente de la variable 'post' el cual pasa de un β_2 de 0.022 a 0.015; además las estimaciones finales de la probabilidad de ventas se verían modificadas por el impacto de las variables climáticas, en particular de la variable *temp* y de la variable *precipitation*. En específico este cambio en las estimaciones se da porque la matriz de variables explicativas ha aumentado la cual sirve de input para calcular los coeficientes individuales. Por otro lado, manteniendo lo demás constante, el hecho de que el comprador y vendedor tengan la probabilidad de comunicarse incrementa la probabilidad de que se concrete una venta en 7 puntos porcentuales ($\beta_1 + \beta_2 + \beta_5$).

Para este modelo también se realizó un boostrap para conocer la estabilidad de los coeficientes. Los resultados se muestran en la Figura 11.

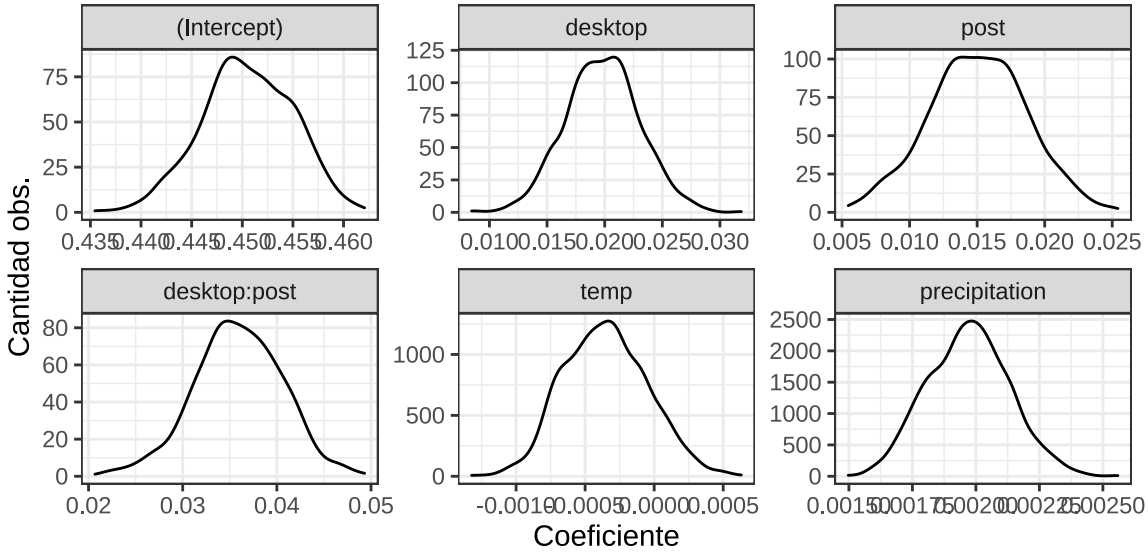

Figure 11: Coeficientes a partir del remuestreo del modelo de regresión incluyendo variable climáticas

La segunda comparación discriminando entre la condición del producto se muestra a continuación. Para el cálculo de los errores se utilizaron para todos los modelos errores robustos dada que no se cumplen los supuestos del modelo lineal.

	Modelo básico	Modelo clima	Modelo nuevo	Modelo usado
(Intercept)	0.448	0.450	0.423	0.471
	[0.443, 0.453]	[0.442, 0.459]	[0.409, 0.436]	[0.460, 0.483]
	s.e. = 0.002	s.e. = 0.004	s.e. = 0.007	s.e. = 0.006
	t = 190.730	t = 100.601	t = 62.129	t = 79.472
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
desktop	0.020	0.020	0.005	0.030
	[0.014, 0.026]	[0.014, 0.026]	[−0.005, 0.014]	[0.021, 0.038]
	s.e. = 0.003	s.e. = 0.003	s.e. = 0.005	s.e. = 0.004
	t = 6.272	t = 6.272	t = 1.004	t = 7.035
	p = 0.000	p = 0.000	p = 0.315	p = 0.000
post	0.022	0.015	0.008	0.020
	[0.016, 0.029]	[0.008, 0.022]	[−0.002, 0.019]	[0.011, 0.029]
	s.e. = 0.003	s.e. = 0.004	s.e. = 0.005	s.e. = 0.005
	t = 6.679	t = 4.193	t = 1.505	t = 4.217
	p = 0.000	p = 0.000	p = 0.132	p = 0.000
desktop × post	0.036	0.036	0.042	0.031
	[0.027, 0.045]	[0.027, 0.045]	[0.029, 0.055]	[0.019, 0.042]
	s.e. = 0.004	s.e. = 0.004	s.e. = 0.007	s.e. = 0.006
	t = 8.000	t = 7.978	t = 6.151	t = 5.149
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
precipitation		0.002	0.002	0.002
		[0.002, 0.002]	[0.001, 0.002]	[0.002, 0.003]
		s.e. = 0.000	s.e. = 0.000	s.e. = 0.000
		t = 12.414	t = 6.917	t = 10.580
		p = 0.000	p = 0.000	p = 0.000
temp		0.000	0.000	−0.001
		[−0.001, 0.000]	[−0.001, 0.001]	[−0.001, 0.000]
		s.e. = 0.000	s.e. = 0.000	s.e. = 0.000
		t = −1.171	t = 0.105	t = −1.570
		p = 0.241	p = 0.917	p = 0.117
Num.Obs.	200 000	200 000	85 150	114 850
R2	0.004	0.004	0.003	0.005
R2 Adj.	0.003	0.004	0.003	0.005
AIC	289 303.1	289 152.9	122 380.6	166 124.5
BIC	289 354.2	289 224.4	122 446.0	166 192.0
Log.Lik.	−144 646.570	−144 569.462	−61 183.289	−83 055.238
F	234.527	172.263		
Std.Errors	Robust	Robust	Robust	Robust

Para productos Nuevos:

Las estimaciones de los coeficientes para el caso de productos nuevos cambian sobre todo en el coeficiente de la variable 'post' el cual pasa de un β_2 de 0.0156 a 0.081; además las

estimaciones finales de la probabilidad de ventas se verían modificadas por el impacto de las variables climáticas, en particular de la variable 'temp' y de la variable 'precipitation'. En específico este cambio en las estimaciones se da porque la matriz de variables explicativas ha aumentado la cual sirve de input para calcular los coeficientes individuales. Asimismo, estos cambios se verían impactados porque en este escenario las variables 'desktop', 'post' y 'temp' resultan estadísticamente no significativos. Por otro lado, manteniendo lo demás constante, el hecho de que el comprador y vendedor tengan la probabilidad de comunicarse incrementa la probabilidad de que se concrete una venta en 7 puntos porcentuales ($\beta_1 + \beta_2 + \beta_5$).

Para productos Usados:

Las estimaciones de los coeficientes para el caso de productos usados cambian sobre todo en el coeficiente de la variable ´post' el cual pasa de 0.027 a 0.020; además las estimaciones finales de la probabilidad de ventas se verían modificadas por el impacto de las variables climáticas, en particular de la variable 'temp' y de la variable 'precipitation'. En específico este cambio en las estimaciones se da porque la matriz de variables explicativas ha aumentado la cual sirve de input para calcular los coeficientes individuales. Asimismo, estos cambios se verían impactados porque en este escenario la variable 'temp' resulta estadísticamente no significativo. Por otro lado, manteniendo lo demás constante, el hecho de que el comprador y vendedor tengan la probabilidad de comunicarse incrementa la probabilidad de que se concrete una venta en 8 puntos porcentuales ($\beta_1 + \beta_2 + \beta_5$).

3.2 Bootstrap manual

```
# ------------------------------------------------------------------------- #
# Paso 9: Bootstrap manual ----
# ------------------------------------------------------------------------- #

# Definir semilla
set.seed(1234)

combinaciones <- data %>%
  dplyr::select(desktop, post) %>%
  dplyr::distinct() %>%
  dplyr::arrange(desktop) %>%
  dplyr::mutate(seed = runif(4, min = 1000, max = 9999))

B = 1000 # Cantidad de repeticiones

bootstrap_media_manual <- purrr::map2_dfr(
  .x = combinaciones$desktop,
  .y = combinaciones$post,
  .f = function(aplicacion, momento) {
```

```
    # Semilla para cada iteracion
    # Se filtran cada una de las combinaciones deseadas
    # de aplicación y momento
    combinaciones %>%
      dplyr::filter(desktop == aplicacion, post == momento) %>%
      dplyr::pull(seed) %>% # Se coloca una semilla para asegurar
                            # la reproducibilidad
      set.seed()

    # Seleccion y filtrado de la variable de interes
    ventas <- data %>%
      dplyr::filter(desktop == aplicacion, post == momento) %>%
      dplyr::pull(itemsold)

    results = c() # Vector para guardar resultados

    # Dentro del for se iteran B veces seleccionado distintas muestras del
    # vector de ventas con reposición para asegurar un n constante
    # sobre cada iteración
    # Luego se calcula la media de cada vector y se guarda en un objeto
    for(b in 1:B){
      # Remuestreo de los datos
      bootSample = sample(ventas, size=length(ventas), replace=TRUE)
      thetaHat = mean(bootSample) # Calculo de la media de la muestra
      results[b] = thetaHat # Guardar resultados
    }

    # Devuelve los resultados con cada media para cada una de las combinaciones
    return(data.frame(desktop = aplicacion, post = momento, media = results))
  }
)

# Calculo de los intervalos de confianza
bootstrap_intervalos_manual <- bootstrap_media_manual %>%
  # Se agrupan los en función de las combinaciones deseadas
  dplyr::group_by(desktop, post) %>%
  # Se oredenan las medias estimadas en el paso anterior
  # de menor a mayor para cada uno de los grupos
  dplyr::arrange(media) %>%
  # Sobre los grupos se seleccionan los valores que están
  # en la vecindad del percentil 2.5 y 97.5, los límites del
  # intervalo de confianza del 95%
  dplyr::summarise(low.int = map_dbl(round(0.025*B), ~ media[.x]),
                   upp.int = map_dbl(round((1 - 0.025)*B), ~ media[.x])) %>%
  # Se desagrupan los resultados
```

```
  dplyr::ungroup() %>%
  # Corrección de nombres para visualizar
  dplyr::mutate(desktop = if_else(desktop == 1, 'Desktop', 'Móvil'),
                post = if_else(post == 1, 'Post', 'Pre'))

# Evaluación de los resultados
bootstrap_media_manual %>%
  dplyr::group_by(desktop, post) %>%
  dplyr::summarise(media = mean(media))
```

La salida del bootstrap manual del caso anterior se muestra en la Figura 12.

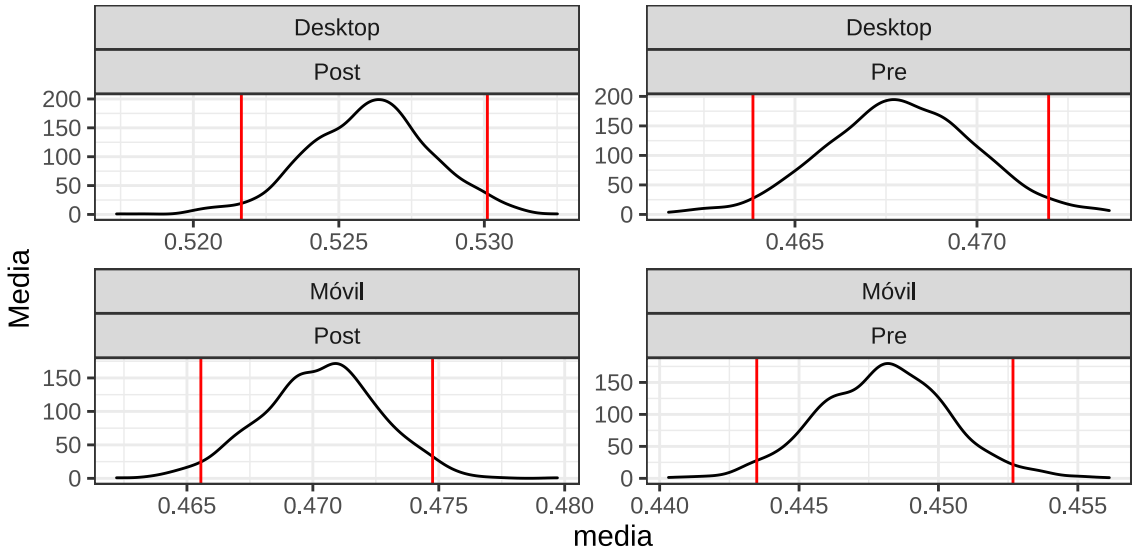

Figure 12: Medias e intervalos de confianza a partir del bootstrap manual.

4 Código utilizado

A continuación se muestra el código completo para la generación es este reporte.

```
# ------------------------------------------------------------------------ #
# Paso 0: Limpieza del espacio de trabajo ----
# ------------------------------------------------------------------------ #

# Eliminar objetos y limpiar memoria
rm(list = ls()); gc()

# ------------------------------------------------------------------------
```

```r
# ------------------------------------------------------------------------- #
# Paso 1: Cargar paquetes necesarios ----
# ------------------------------------------------------------------------- #

Sys.setenv(TZ = "UTC")
list.of.packages <- c("dplyr", "magrittr", "ggplot2", "purrr", "magrittr",
                      "lubridate", "tidyr", "readr", "funModeling", "kableExtra",
                      "dlookr", "pwr", "gstools", "equatiomatic", "margins",
                      "rsample")
for (pack in list.of.packages) {
  if (!require(pack, character.only = TRUE)) {
    stop(paste0("Paquete no encontrado: ", pack))
  }
}

options(bitmapType="cairo")

rm(pack); gc()

# ------------------------------------------------------------------------

# ------------------------------------------------------------------------- #
# Paso 2: Lectura del dataset ----
# ------------------------------------------------------------------------- #

data <- readr::read_delim("./data/ebay_data.csv", delim = ';') %>%
  # Corregir formato de fecha
  dplyr::mutate(date = lubridate::dmy(date))

# ------------------------------------------------------------------------

# ------------------------------------------------------------------------- #
# Paso 3: Exploración del dataset ----
# ------------------------------------------------------------------------- #

# Tipos de variables
tipos_datos <- dlookr::diagnose(data)

# Diagnóstico numerico
diagnostico_variables_numericas <- dlookr::diagnose_numeric(data)

# Diagnóstico categorico
diagnostico_variables_categoricas <- dlookr::diagnose_category(data %>%
                                                               dplyr::select(category)
```

```r
## Análisis univariado

# Cantidad de ventas por plataforma
ventas_plataforma <- ggplot2::ggplot(data = data, ggplot2::aes(x = as.factor(desktop))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Plataforma") + ggplot2::ylab("Cantidad de operaciones") +
  ggplot2::scale_x_discrete(label = c('Móvil', 'Desktop'))
ventas_plataforma_concretadas <- ggplot2::ggplot(data = data %>%
                                                   dplyr::filter(itemsold == 1),
                                                 ggplot2::aes(x = as.factor(desktop))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Plataforma") + ggplot2::ylab("Cantidad de ventas") +
  ggplot2::scale_x_discrete(label = c('Móvil', 'Desktop'))


# Cantidad de ventas por momento
ventas_momento <- ggplot2::ggplot(data = data, ggplot2::aes(x = as.factor(post))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Momento de compra") + ggplot2::ylab("Cantidad de operaciones") +
  ggplot2::scale_x_discrete(label = c('Pre', 'Post'))
ventas_momento_concretadas <- ggplot2::ggplot(data = data %>%
                                                dplyr::filter(itemsold == 1),
                                              ggplot2::aes(x = as.factor(post))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Momento de compra") + ggplot2::ylab("Cantidad de ventas") +
  ggplot2::scale_x_discrete(label = c('Pre', 'Post'))

# Cantidad de ventas por categoria
ventas_categoria <- ggplot2::ggplot(data = data, ggplot2::aes(x = category)) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Categorías") + ggplot2::ylab("Cantidad de ventas")

# Cantidad de mensajes intercambiados
ventas_mensajes <- ggplot2::ggplot(data = data, ggplot2::aes(x = as.factor(message))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Mensajes enviados?") + ggplot2::ylab("Cantidad de ventas") +
  ggplot2::scale_x_discrete(label = c('No', 'Si'))
```

```r
# Cantidad de condicion del producto
condicion_venta <- ggplot2::ggplot(data = data, ggplot2::aes(x = as.factor(condition))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Condición \ndel producto") + ggplot2::ylab("Cantidad de operaciones") +
  ggplot2::scale_x_discrete(label = c('Usado', 'Nuevo'))
condicion_venta_concretada <- ggplot2::ggplot(data = data %>%
                                                dplyr::filter(itemsold == 1),
                                              ggplot2::aes(x = as.factor(condition))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Condición \ndel producto") + ggplot2::ylab("Cantidad de ventas") +
  ggplot2::scale_x_discrete(label = c('Usado', 'Nuevo'))

# Densidad del precio escala natural
densidad_precio <- ggplot2::ggplot(data = data, ggplot2::aes(x = askingprice)) +
  ggplot2::geom_density() +
  ggplot2::geom_rug() +
  ggplot2::xlab("Asking price") + ggplot2::ylab("Densidad") +
  ggplot2::theme_bw()
# Densidad del precio escala log10
densidad_precio_log <-ggplot2::ggplot(data = data, ggplot2::aes(x = askingprice)) +
  ggplot2::geom_density() +
  ggplot2::geom_rug(alpha = 0.2) +
  ggplot2::scale_x_log10() +
  ggplot2::xlab("Asking price") + ggplot2::ylab("Densidad") +
  ggplot2::theme_bw()

## Efecto de las condiciones meteorológicas

# Temperatura
pctiles <- seq(0, 1, 0.20)

temperatura_venta <- data %>%
  mutate(percentile = gtools::quantcut(temp, q=seq(0, 1, by=0.2)))  %>%
  ggplot2::ggplot(data = ., ggplot2::aes(x = as.factor(percentile))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Percentil de temperatura") + ggplot2::ylab("Cantidad de operaciones")
temperatura_venta_concretada <- data %>%
  mutate(percentile = gtools::quantcut(temp, q=seq(0, 1, by=0.2)))  %>%
  dplyr::filter(itemsold == 1) %>%
  ggplot2::ggplot(data = .,
                  ggplot2::aes(x = as.factor(percentile))) +
  ggplot2::geom_bar() +
```

```r
  ggplot2::theme_bw() +
  ggplot2::xlab("Percentil de temperatura") + ggplot2::ylab("Cantidad de ventas")

# Precipitación
precipitacion_venta <- data %>%
  mutate(dia_lluvioso = if_else(precipitation > 0.5, "Lluvioso", "Seco"))  %>%
  ggplot2::ggplot(data = ., ggplot2::aes(x = as.factor(dia_lluvioso))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Día lluvioso") + ggplot2::ylab("Cantidad de operaciones")
precipitacion_venta_concretada <- data %>%
  mutate(dia_lluvioso = if_else(precipitation > 0.5, "Lluvioso", "Seco"))  %>%
  dplyr::filter(itemsold == 1) %>%
  ggplot2::ggplot(data = ., ggplot2::aes(x = as.factor(dia_lluvioso))) +
  ggplot2::geom_bar() +
  ggplot2::theme_bw() +
  ggplot2::xlab("Día lluvioso") + ggplot2::ylab("Cantidad de ventas")


# Diagnóstico de outliers
diagnostico_outliers <- dlookr::diagnose_outlier(data %>%
                                                   dplyr::select(askingprice, temp, prec

data %>%
  dplyr::select(askingprice) %>%
  dplyr::mutate(lower.bound = median(askingprice) - 3 * mad(askingprice, constant = 1),
                upper.bound = median(askingprice) + 3 * mad(askingprice, constant = 1),
                outlier = if_else(askingprice >= upper.bound | askingprice <= lower.boun
  ggplot2::ggplot(data = ., ggplot2::aes(x = askingprice, fill = outlier)) +
  ggplot2::geom_density()

# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------------ #
# Paso 4: Creación de intervalos de confianza ----
# ------------------------------------------------------------------------------ #

# Funcion para el calculo de la media
meanfun <- function(data, i){
  d <- data[i]
  return(mean(d))
}

# Definir semilla
set.seed(1234)
```

```r
combinaciones <- data %>%
  dplyr::select(desktop, post) %>%
  dplyr::distinct() %>%
  dplyr::arrange(desktop) %>%
  dplyr::mutate(seed = runif(4, min = 1000, max = 9999))

bootstrap_media <- purrr::map2(
  .x = combinaciones$desktop,
  .y = combinaciones$post,
  .f = function(aplicacion, momento) {


    # Semilla para cada iteracion
    combinaciones %>%
      dplyr::filter(desktop == aplicacion, post == momento) %>%
      dplyr::pull(seed) %>%
      set.seed()

    # Seleccion y filtrado de la variable de interes
    ventas <- data %>%
      dplyr::filter(desktop == aplicacion, post == momento) %>%
      dplyr::pull(itemsold)

    # Remuestreo de la media de ventas
    boot_media <- boot::boot(data = ventas, statistic = meanfun, R = 1000)

    # Creacion de variable para nombrar la lista resultante
    nombre_lista <- paste("Desktop:" , aplicacion, "| Post:", momento)
    # Crear objeto para guardar resultados
    resultados <- list(boot_media)
    # Renombrar objeto resultado
    resultados %<>% purrr::set_names(nombre_lista)

  }
) %>% unlist(., recursive = FALSE) # Eliminar un nivel de la lista.


bootstrap_media_conf <- purrr::map_dfr(
  .x = unique(names(bootstrap_media)),
  .f = function(combinacion) {

    boot_media <- bootstrap_media[[combinacion]]

    broom::tidy(boot_media, conf.int = T) %>%
```

```
      dplyr::mutate(desktop = stringr::str_sub(combinacion, start = 10, end = 10),
                    post = stringr::str_sub(combinacion, start = -1, end = -1)) %>%
      dplyr::select(desktop, post, media = statistic, sesgo = bias, error = std.error,
                    conf.inf = conf.low, conf.sup = conf.high)

  }
)

intervalos_confianza_plataforma_momento <- ggplot(data = bootstrap_media_conf, ggplot2::
  ggplot2::geom_point() +
  ggplot2::scale_x_discrete("Plataforma", labels = c("Móvil", "Escritorio")) +
  ggplot2::scale_color_discrete("Momento", labels = c("Pre", "Post")) +
  ggplot2::geom_errorbar(aes(ymin=conf.inf, ymax=conf.sup), width = .1) +
  ggplot2::theme_bw() +
  ggplot2::xlab("Plataforma") + ggplot2::ylab("Media")
# ------------------------------------------------------------------------------

# ---------------------------------------------------------------------------- #
# Paso 5: Evaluación empírica de ventas + potencia del test ----
# ---------------------------------------------------------------------------- #

ventas <- data %>%
  dplyr::filter(desktop == 1,
                post == 1,
                condition == 1) %>%
  dplyr::pull(itemsold)

# Test de hipotesis sobre la media
prueba_media <- prop.test(x = sum(ventas), p = 0.5, n = length(ventas),
                          alternative = "greater", correct = FALSE)

prueba_media_tabla <- broom::tidy(prueba_media)

# Definición de funcion para la creación de secuencias logaritmicas
seq_log <- function(from = 1, to = 100000, by = 1, length.out = log10(to/from)+1) {
  tmp <- exp(seq(log(from), log(to), length.out = length.out))
  tmp[seq(1, length(tmp), by)]
}

# Definicion de funcion para el calculo de la potencia del test
potencia_prueba_test <- function(null.pi, true.pi, n, alpha = 0.05, alternative = "not e
  # TO DO: Faltan incluir controles de ejecución

  # Seleccion del tipo de prueba a realizar: cola izquierda o derecha
  # o a dos colas
```

```
  # A partir de del tipo de prueba se obtiene el z critico
  z.critico = switch(alternative, "less" = qnorm(alpha),
                     "greater" = qnorm(1-alpha), qnorm(1-alpha/2))

  # Calculo del cuantil para el calculo de la probabilidad
  cuantil.una.cola = (z.critico*sqrt(null.pi*(1-null.pi)/n) + null.pi - true.pi) / sqrt(

  # Corrección para dos colas
  if (alternative == "not equal") {
    z.critico = qnorm(alpha/2)
    # Calculo del cuantil para el calculo de la probabilidad
    cuantil.dos.colas = (z.critico*sqrt(null.pi*(1-null.pi)/n) + null.pi - true.pi) / sq
  }

  # Calculo de la potencia
  potencia = switch(alternative,
                    "less" = pnorm(cuantil.una.cola),
                    "greater" = 1-pnorm(cuantil.una.cola),
                    pnorm(cuantil.dos.colas) + (1-pnorm(cuantil.una.cola))
  )
  # Devolver potencia
  return(potencia)
}

#potencia_prueba_test(null.pi=0.45, true.pi=c(.5, .6, .8), n= c(10), alternative="greate


# Evaluar la potencia del test para distintas
potencia_test <- purrr::map_dfr(
  .x = seq_log(from = 100, to = 100000),
  .f = function(n) {

    # Vector de valores de pi
    pistar <- seq(from = 0.5, to=.52, by=0.001)

    potencia <- potencia_prueba_test(null.pi = 0.5, true.pi = pistar, n = n,
                      alpha = 0.05, alternative = 'greater')

    data.frame(tamano_muestral = factor(n),
               pistar = pistar,
               potencia = potencia)

  }
)
```

```
potencia_prueba <- ggplot2::ggplot(data = potencia_test, ggplot2::aes(x = pistar, y = po
  ggplot2::geom_line() +
  ggplot2::scale_color_discrete(name = 'Tamaño muestral') +
  ggplot2::theme_bw() +
  ggplot2::xlab(bquote(pi^'*')) + ggplot2::ylab("Potencia")


# Prueba de potencia para una distribucion binomial
#pwr::pwr.p.test(n = 5000, h = 0.5, alternative = "greater")


# ------------------------------------------------------------------------------

# ---------------------------------------------------------------------------- #
# Paso 6: Modelo de regresión lineal básico ----
# ---------------------------------------------------------------------------- #

# Independiente de la condicion del producto

# Formula del modelo lineal
formula_modelo <- formula("itemsold~desktop + post + desktop * post")

# Regresion por MCO
modelo_regresion_basico <- lm(formula_modelo, data = data)

# Promediode los residuos == 0
mean(modelo_regresion_basico$residuals)

# Regresion por MCO haciendo bootstrap para conocer la distribución de
# los coeficientes del modelo

# Semilla para el remuestreo
set.seed(1234)

# Para el remuestreo se utiliza el paquete rsample. Forma parte de la suite
# de tidyverse y permite utilizar la funcionalidad de los paquetes relacionados

# Se crean N remuestreos del dataset original
# TO DO: Elegir in R lo suficientemente grande, se usa 100 solo para pruebas
bootstrapped_samples_basico <- rsample::bootstraps(data, times = 1000)


# Definición de función para ajustar el modelo lineal
lm_coefs <- function(splits, ...) {
```

```
  # se `analysis` para extraer el data frame correspondiente a
  # cada muestra
  lm(..., data = rsample::analysis(splits)) %>%
    broom::tidy() # tidy permite extraer los coeficientes ajustados
}

# Se itera sobre cada una de los remuestreos para el ajuste del modelo
# lineal y la extracción de los coeficientes
bootstrapped_samples_basico$model <- purrr::map(
                          .x = bootstrapped_samples_basico$splits,
                          .f = lm_coefs, formula_modelo)

# Extraer los coeficientes ajustados para cada muestra y convertir en un
# data frame para poder graficar
lm_coef_basico <- bootstrapped_samples_basico %>%
  dplyr::select(-splits) %>%
  # Apilar los tibbles en la variable model
  tidyr::unnest(model) %>%
  # Seleccionar las variables de interes
  dplyr::select(id, term, estimate, std.error, statistic,  p.value) %>%
  dplyr::mutate(term = factor(term, levels = c('(Intercept)', 'desktop', 'post', 'deskto

# Intervalos percentiles
p_ints_basico <- rsample::int_pctl(bootstrapped_samples_basico, model)

histograma_coeficientes_basico <- ggplot2::ggplot(data = lm_coef_basico, ggplot2::aes(x
  ggplot2::geom_density() +
  ggplot2::facet_wrap(~term, scales = 'free') +
  ggplot2::geom_vline(data = p_ints_basico, aes(xintercept = .lower), col = "red") +
  ggplot2::geom_vline(data = p_ints_basico, aes(xintercept = .upper), col = "red") +
  ggplot2::theme_bw() +
  ggplot2::xlab("Coeficiente") + ggplot2::ylab("Cantidad")


relacion_coeficientes <- lm_coef_basico %>%
  dplyr::select(id, term, estimate) %>%
  # Put different parameters in columns
  tidyr::spread(term, estimate) %>%
  # Keep only numeric columns
  dplyr::select(-id) %>%
  GGally::ggscatmat(alpha = .25) +
  ggplot2::theme_bw() +
  ggplot2::xlab("Valor del estimador (eje x)") +
  ggplot2::ylab("Valor del estimador (eje y)")
```

```
# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------- #
# Paso 7: Modelo de regresión lineal por condicion de producto ----
# ------------------------------------------------------------------------- #

# Formula del modelo lineal
formula_modelo_condicion <- formula("itemsold~desktop + post + desktop * post")

# Discriminando entre producto nuevo y usado
modelo_regresion_condicion <- purrr::map(
  .x = unique(data$condition),
  .f = function(condicion) {

    # Filtrar datos por condicion
    data_condicion <- data %>%
      dplyr::filter(condition == condicion)

    modelo_regresion <- lm(formula_modelo_condicion, data = data_condicion)

    #broom::tidy(modelo_regresion) %>%
    #  dplyr::mutate(signif = p.value < 0.05,
    #    condition = condicion)

    # Creacion de variable para nombrar la lista resultante
    nombre_lista <- paste("Condicion:" , condicion)
    # Crear objeto para guardar resultados
    resultados <- list(modelo_regresion)
    # Renombrar objeto resultado
    resultados %<>% purrr::set_names(nombre_lista)

  }
) %>% unlist(., recursive = FALSE) # Eliminar un nivel de la lista.

# Remuestreo de coeficientes discriminando entre producto nuevo y usado
bootstrap_model_condition <- purrr::map_dfr(
  .x = unique(data$condition),
  .f = function(condicion) {

    set.seed(condicion)
    bt_resamples <-  rsample::bootstraps(data %>%
                                dplyr::filter(condition == condicion), times = 1000)

    bt_resamples$model <- purrr::map(.x = bt_resamples$splits,
                              .f = lm_coefs,
```

```
                                formula_modelo_condicion)

    lm_coef <-
      bt_resamples %>%
      dplyr::select(-splits) %>%
      # Turn it into a tibble by stacking the `models` col
      unnest() %>%
      dplyr::mutate(condition = condicion) %>%
      # Get rid of unneeded columns
      dplyr::select(id, condition, term, estimate, std.error, statistic,  p.value)

  }
)

histograma_coeficientes_condicion <- ggplot2::ggplot(data = bootstrap_model_condition,
                                                     ggplot2::aes(x = estimate, fill = as.factor(c
  ggplot2::geom_density(alpha = 0.4) +
  ggplot2::scale_fill_discrete(name = 'Condición') +
  ggplot2::facet_wrap(.~term, scales = 'free') +
  ggplot2::theme_bw() +
  ggplot2::xlab("Coeficiente") + ggplot2::ylab("Cantidad obs.") +
  ggplot2::theme(legend.position = 'bottom')


bootstrap_model_condition %>%
  dplyr::group_by(condition, term) %>%
  dplyr::summarise(estimate = mean(estimate))

# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------- #
# Paso 8: Modelo de regresión lineal con condiciones climaticas ----
# ------------------------------------------------------------------------- #

# Incorporacion de variables climáticas

# Formula del modelo lineal
formula_modelo_clima <- formula("itemsold~desktop + post + desktop * post + precipitatic

# Regresion por MCO
modelo_regresion_clima <- lm(formula_modelo_clima, data = data)

# Se crean N remuestreos del dataset original
# TO DO: Elegir in R lo suficientemente grande, se usa 100 solo para pruebas
bootstrapped_samples_clima <- rsample::bootstraps(data, times = 1000)
```

```
# Se itera sobre cada una de los remuestreos para el ajuste del modelo
# lineal y la extracción de los coeficientes
bootstrapped_samples_clima$model <- purrr::map(.x = bootstrapped_samples_clima$splits,
                                               .f = lm_coefs, formula_modelo_clima)

# Extraer los coeficientes ajustados para cada muestra y convertir en un
# data frame para poder graficar
lm_coef_clima <- bootstrapped_samples_clima %>%
  dplyr::select(-splits) %>%
  # Apilar los tibbles en la variable model
  tidyr::unnest(model) %>%
  # Seleccionar las variables de interes
  dplyr::select(id, term, estimate, std.error, statistic,  p.value) %>%
  dplyr::mutate(term = factor(term, levels = c('(Intercept)', 'desktop', 'post', 'desktc


histograma_coeficientes_clima <- ggplot2::ggplot(data = lm_coef_clima,
                                                 ggplot2::aes(x = estimate)) +
  ggplot2::geom_density(alpha = 0.4) +
  ggplot2::scale_fill_discrete(name = 'Condición') +
  ggplot2::facet_wrap(.~term, scales = 'free') +
  ggplot2::theme_bw() +
  ggplot2::xlab("Coeficiente") + ggplot2::ylab("Cantidad obs.") +
  ggplot2::theme(legend.position = 'bottom')



# Discriminando entre producto nuevo y usado
modelo_regresion_condicion_clima <- purrr::map(
  .x = unique(data$condition),
  .f = function(condicion) {

    # Filtrar datos por condicion
    data_condicion <- data %>%
      dplyr::filter(condition == condicion)

    modelo_regresion <- lm(formula_modelo_clima, data = data_condicion)

    #broom::tidy(modelo_regresion) %>%
    #  dplyr::mutate(signif = p.value < 0.05,
    #                condition = condicion)

    # Creacion de variable para nombrar la lista resultante
```

```
    nombre_lista <- paste("Condicion:" , condicion)
    # Crear objeto para guardar resultados
    resultados <- list(modelo_regresion)
    # Renombrar objeto resultado
    resultados %<>% purrr::set_names(nombre_lista)

  }
) %>% unlist(., recursive = FALSE) # Eliminar un nivel de la lista.
# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------------ #
# Paso 9: Bootstrap manual ----
# ------------------------------------------------------------------------------ #

# Definir semilla
set.seed(1234)

combinaciones <- data %>%
  dplyr::select(desktop, post) %>%
  dplyr::distinct() %>%
  dplyr::arrange(desktop) %>%
  dplyr::mutate(seed = runif(4, min = 1000, max = 9999))

B = 1000 # Cantidad de repeticiones

bootstrap_media_manual <- purrr::map2_dfr(
  .x = combinaciones$desktop,
  .y = combinaciones$post,
  .f = function(aplicacion, momento) {

    # Semilla para cada iteracion
    # Se filtran cada una de las combinaciones deseadas
    # de aplicación y momento
    combinaciones %>%
      dplyr::filter(desktop == aplicacion, post == momento) %>%
      dplyr::pull(seed) %>% # Se coloca una semilla para asegurar
                            # la reproducibilidad
      set.seed()

    # Seleccion y filtrado de la variable de interes
    ventas <- data %>%
      dplyr::filter(desktop == aplicacion, post == momento) %>%
      dplyr::pull(itemsold)

    results = c() # Vector para guardar resultados
```

```
    # Dentro del for se iteran B veces seleccionado distintas muestras del
    # vector de ventas con reposición para asegurar un n constante
    # sobre cada iteración
    # Luego se calcula la media de cada vector y se guarda en un objeto
    for(b in 1:B){
      bootSample = sample(ventas, size=length(ventas), replace=TRUE) # Remuestreo de los
      thetaHat = mean(bootSample) # Calculo de la media de la muestra
      results[b] = thetaHat # Guardar resultados
    }

    # Devuelve los resultados con cada media para cada una de las combinaciones
    return(data.frame(desktop = aplicacion, post = momento, media = results))
  }
)

# Calculo de los intervalos de confianza
bootstrap_intervalos_manual <- bootstrap_media_manual %>%
  # Se agrupan los en función de las combinaciones deseadas
  dplyr::group_by(desktop, post) %>%
  # Se oredenan las medias estimadas en el paso anterior
  # de menor a mayor para cada uno de los grupos
  dplyr::arrange(media) %>%
  # Sobre los grupos se seleccionan los valores que están
  # en la vecindad del percentil 2.5 y 97.5, los límites del
  # intervalo de confianza del 95%
  dplyr::summarise(low.int = map_dbl(round(0.025*B), ~ media[.x]),
                   upp.int = map_dbl(round((1 - 0.025)*B), ~ media[.x])) %>%
  # Se desagrupan los resultados
  dplyr::ungroup() %>%
  # Corrección de nombres para visualizar
  dplyr::mutate(desktop = if_else(desktop == 1, 'Desktop', 'Móvil'),
                post = if_else(post == 1, 'Post', 'Pre'))

# Evaluación de los resultados
bootstrap_media_manual %>%
  dplyr::group_by(desktop, post) %>%
  dplyr::summarise(media = mean(media))


bootstrap_intervalos_manual_plot <- ggplot(data = bootstrap_media_manual %>%
                                             dplyr::mutate(desktop = if_else(desktop ==
                                                           post = if_else(post == 1, 'Po
  ggplot2::geom_density() +
  ggplot2::facet_wrap(desktop~post, scales = 'free') +
```

```
  ggplot2::geom_vline(data = bootstrap_intervalos_manual, ggplot2::aes(xintercept = low.
  ggplot2::geom_vline(data = bootstrap_intervalos_manual, ggplot2::aes(xintercept = upp.

  ggplot2::theme_bw() +
  ggplot2::ylab("Media")
# ------------------------------------------------------------------------------

# ------------------------------------------------------------------------------ #
# ---- Paso n: Generar reporte ----
# ------------------------------------------------------------------------------ #

# Generar informe en PDF

output.dir <- getwd()
rmarkdown::render(
  input = "./trabajo_practico.Rmd",
  output_file = "trabajo_practico.pdf",
  output_dir = output.dir
)

# ------------------------------------------------------------------------------
```