



A/B TESTING

▼ SET THE EXPERIMENT

▼ OBJETIVES

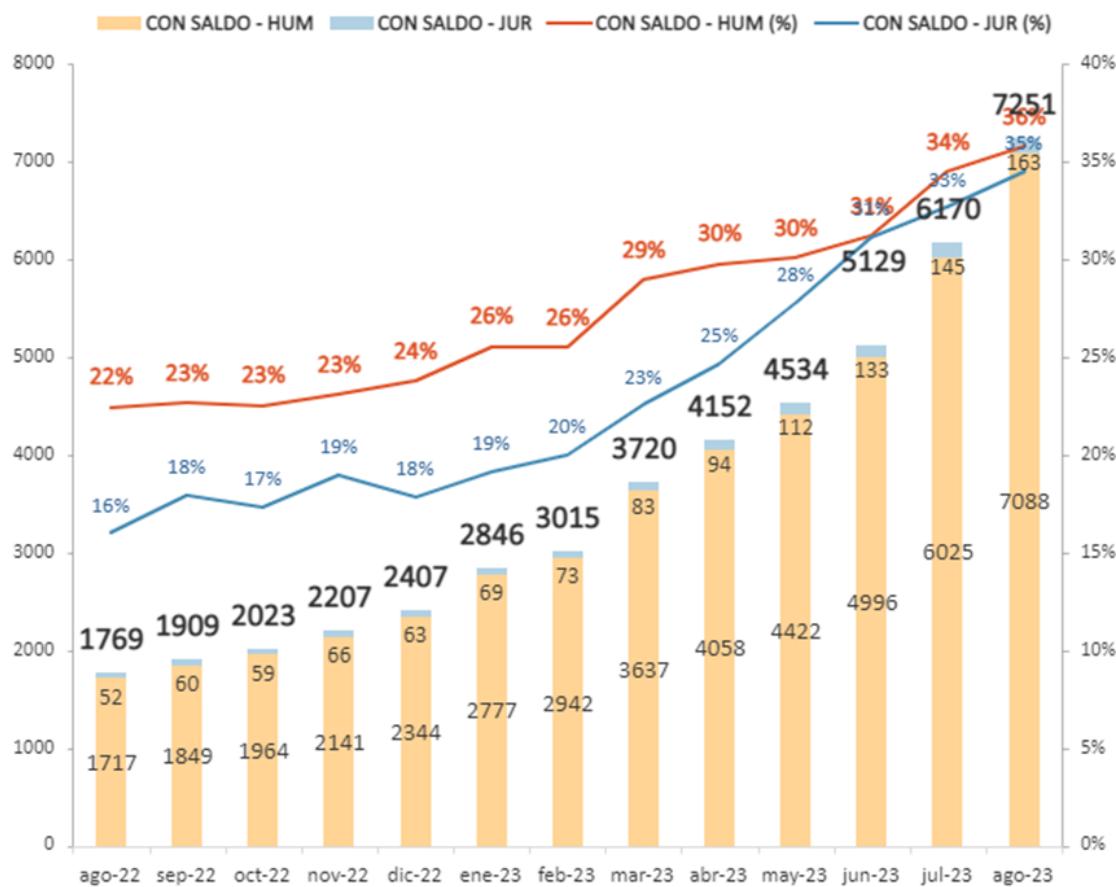
Determine whether is better to communicate to our clients via POP UP or via EMAIL.

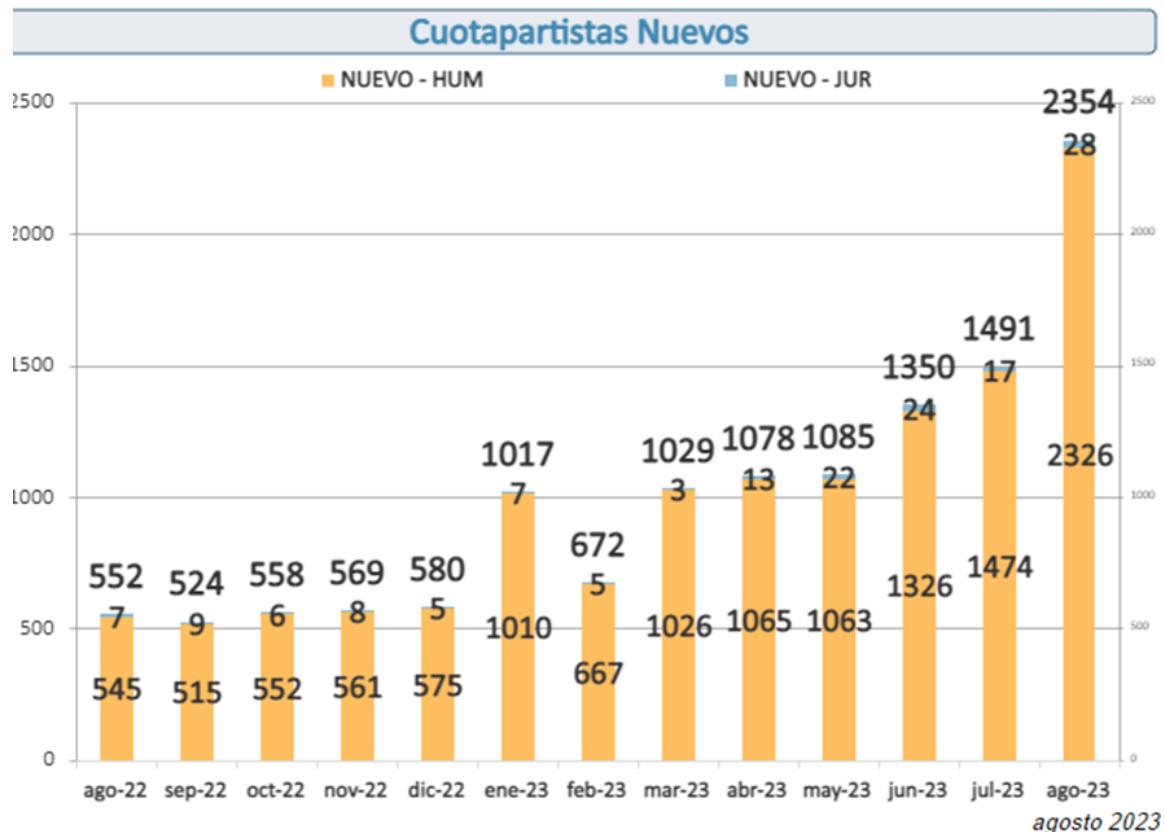
Launch our first A/B TESTING

A key value is to have clear in mind which is the goal of the experiment. If we have many targets, the results and the focus of the process might be complex.

▼ BACKGROUND

Cuotapartistas con Saldo





▼ SUBSET

Invitation to the clients to acquire a position in our mutual funds.

Banco Ciudad - Fondos Comunes de Inversión

Enterate de todas las novedades y beneficios que tenemos para vos.

🌐 <https://www.bancociudad.com.ar/institucional/personas/Inversiones/Fondos%20Comunes#?tab=3>



▼ THE CODE

▼ SELECT SEGMENT

```
#CLIENTS WHO TRANSFERS THEIR INCOMES DURING 1st July to XX Ju
```

SELECT

```
CC.cliente,  
SUM(monto) transfe,
```

```

        MONTH(fechahora) as Mes,
        YEAR(fechahora) as Año
    FROM Datawarehouse.dbo.V_TransaccionesCobis t

    INNER JOIN nexos.dbo.clientescuentas cc
        on t.Dependencia = cc.Dependencia
        and t.Sistema = cc.sistema
        and t.cuenta = cc.cuenta
        and cc.baja is null
        and calidad = '01' --titular
        and orden = 1 --1er titular

    WHERE
        ((year(Fechahora) = 2023 and month(Fechahora) = 7 and day(fechahora) = 1)
        or (year(Fechahora) = 2023 and month(Fechahora) = 6 and day(fechahora) = 1))
        and ((DescCausal like '%mismo%' and DescripcionTransaccion like '%cambio de domicilio%')
        and left(cc.cliente, 2) not in ('88', '85'))

    GROUP BY
        year(fechahora),
        month(fechahora),
        CC.cliente

```

```

# IN ORDER TO SUM THE TRANSFERS MADE DURING BOTH MONTHS. PERHAPS WE CAN USE THE SUM FUNCTION
#Add the month as a new column to the DataFrame
query["Mes"] = 7
#SUM of the total amount per month (days from the month before)
final_result = query.groupby(["Mes", "Año", "cliente"], as_index=False).sum()

```

```

#QUERY TO ENRICH AND FILTERS THE PREVIOUS QUERY
# + INCOMES INFO
# + EMAIL
# + EXLCUDE RETIRED
SELECT
    month(FechaSaldos) as Mes
    ,year(FechaSaldos) as Año

```

```

,cliente
,RazonSocial
,isnull(case when ltrim(rtrim(email)) = '' then NULL
,SaldoCajaHaber as Sueldo
FROM DataMining.dbo.wh_clientes_ultima_enriquecida_1344
WHERE
    TipoDocumento NOT IN ('88' , '85')
    and segmento_planeam <> 'empleados_del_banco'
    and SaldoJubi = 0

GROUP BY
    month(fechasaldos) ,
    year(fechasaldos) ,
    cliente,
    RazonSocial,
    email,
    email_corregido_enriquecido,
    SaldoCajaHaber

```

```

#OTHER IMPORTANT FILTERS.
# TO CHECK THE COMPLETE CODE , I LEFT THE PATH:
#L:\02-Tareas_programadas\Juanma\08-Clientes_transfieren\Expe

EXCLUYO TRASNFERENCIA MENORES a SALARIO BASICO JULIO 2023
df2 = df[df.transfe> 105500]

#DEJO TRANSFERENCIAS MENORES O IGUAL AL INGRESO
df3 = df2[df2.transfe <= df2.Sueldo]

# GENERO COLUMNA % transfe/sueldo
df3[%transfe/ingreso"] = (df3["transfe"]/df3["Sueldo"]).round(2)

#Completo mail en blanco con "sin dato"
df3[email"]=df3[email"].fillna("sin dato")

#FILTRO VALORES @BCBA

```

```

df4 = df3[~df3['email'].str.contains('@bancociudad.com.ar')]

#FILTRO VIS
df5 = df4[~df4['email'].str.contains('@BANCOCIUDAD.COM.AR')]

#FILTRO TRANSFERENCIAS MAYORES O IGUALES A 50%
df6 = df6[df6["%transfe/ingreso"]>=0.5]

#ELIMINO MAIL SIN DATO
df7 = df6[df6["email"]!="sin dato"]

#LEVANTO BASE CON SUSCRIPTORES DE FCI DURANTE JULIO 2023
clientes_fci= pd.read_excel("L:/02-Tareas_programadas/Juanma/")

#ELIMINO LOS QUE CRUZAN CON LA BASE DE FONDOS FCI
df9 = df8[df8["_merge"]!="both"]

```

▼ RANDOMIZE

It's crucial to correctly randomize. In the experiment we have the following variables to randomize

- INCOME
- TRANSFER
- % TRANSFER/INCOME

The other variables identified the client and month

```

# Randomize the order of clients #FRAC 1 mantiene la misma cantidad de clientes
randomized_df = df12.sample(frac=1).reset_index(drop=True)

# Split the randomized DataFrame into 3 equal parts
num_groups = 3
group_size = len(randomized_df) // num_groups

groups = []

#CARGO DATOS DE CADA GRUPO

```

```
for i in range(num_groups):
    start_idx = i * group_size
    end_idx = start_idx + group_size
    group = randomized_df[start_idx:end_idx]
    groups.append(group)
```

▼ Z TEST

- Is a test used to determine if two groups are SIGNIFICANTLY different one from each other.



REQUIREMENTS:

1. Variables should be continuous. In this case, ratio conversions
2. Normally distributed
3. Have similar size the 2 groups

Online calculation

<https://abtestguide.com/calc/>

Z TEST - Code

```
import pandas as pd
import numpy as np
from statsmodels.stats.proportion import proportions_ztest
#pip install statsmodels

#Paso 1 Definición de hipotesis. OK

#Paso 2 Definición de nivel de confianza. 95%. LINK: https://
alpha= 0.05
```

```

#Paso 3 Calculo de valor de P según test estadístico elegido
#valores
conversion_hb= 3
muestra_hb= 2347

conversion_email= 1
muestra_email= 1982

#conversión a array de datos
array_conversion = np.array([conversion_hb, conversion_email])
array_muestra = np.array([muestra_hb,muestra_email])

p_valor = proportions_ztest(count=array_conversion, nobs=array_muestra)
print(f"{p_valor:.3f}")

if p_valor <= alpha:
    print("Se rechaza la hipotesis nula")
else:
    print("Se acepta la hipotesis nula")

```

▼ TIMELINE + DAY BY DAY ASPECTS

The teams diagram

Orchestration - Schedule + Trigger

- 1- Reunión de kick-off con todos los participantes
- 2- Definir características de la base y ID a utilizar para poder realizar seguimiento y balanceo de los samples
- 3- Establecer y agendar fecha de envío de comunicación
- 4- Confirmar envío de comunicación por parte del equipo correspondiente
- 5- Evaluar tiempo necesario para levantar resultados

6- Levantar resultados de experimentación por ICOMM

7- Realizar test

▼ RESULTS

▼ FIRST ROUND

1. HB

4/2452 0.16% (Conversions/Group)

3/2347 0.12% (Conversion/Group) Last access after the experiment performs

2. MAIL

10/2465 0.5% (Conversions/Group)

9/1959 0.45% (Conversions/Sent emails)

1/408 0.24% (Conversions/Opened emails)

3. CONTROL

5/2363 0.21% (Conversions/Group)

Path to the file with the results

L:\28-Experimentos con
Promociones\Experimento_N1\Experimento_N1_analisis.xlsx

▼ SECOND ROUND

1. HB

19/3330 0.57% (Conversions/Group)

19/3174 0.598% (Conversion/Group) Last access after the experiment performs

2. MAIL

23/3334 0.69% (Conversions/Group)

20/2754 0.73% (Sent emails/Group)

9/557 1,61% (Opened emails/Group)

3. CONTROL

22/3334 **0.66% (Conversions/Group)**

Path to the file with the results

L:\28-Experimentos con Promociones\Experimento_N2\Experimento_N2_analisis.xlsx

Key results

1. Low conversion rate
2. No difference between pop up and email
3. Few customers open emails

▼ FIXED TERM DEPOSITS

1. 31/07/2023 campaign
803 new shareholders from 109.762 **0,7%**
2. 30/08/2023 campaign
1418 new shareholders from 123.654 clients. **1,14%**

▼ MEETINGS

1. 1st meeting with LUCA:

- a. **What if we extend the analysis time**

- 1st round - Experiment:

From 19 shareholders to 59!!

Add groups

We extend the analysis time from 1-14/8 to 1/8 to 13/9:

- 2nd round - Experiment:

From 64 shareholders to ...

We didn't receive information about subscriptions till 26/09/2023.

- b. **Behavior of fixed term funds clients**

1. 30/08/2023 campaign

3.072 new shareholders from 123.654 clients. **2,48%**

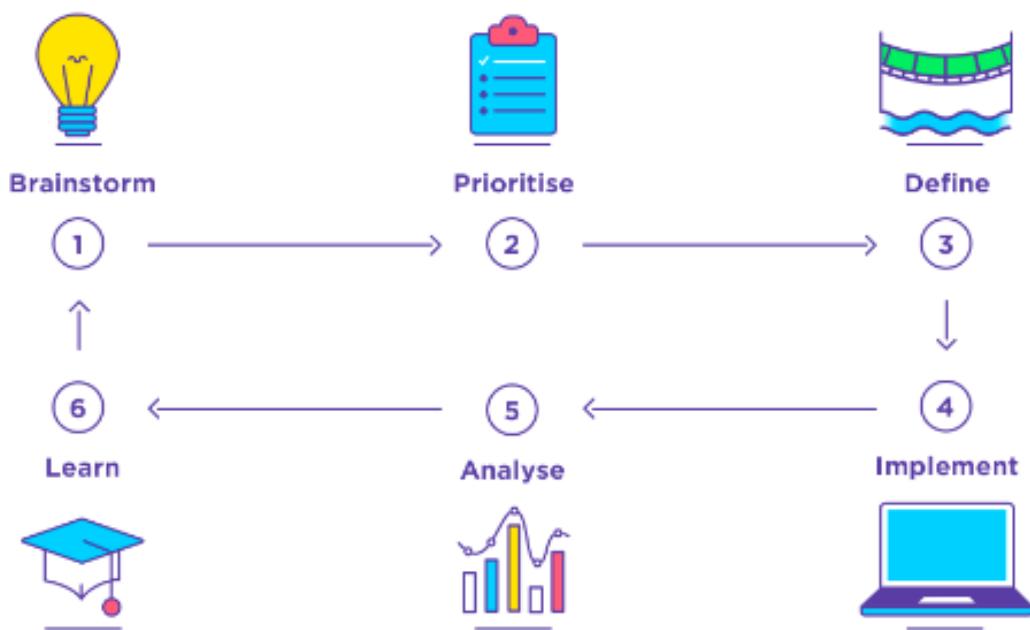
▼ CONCLUSIONS

- We have launched the first experiment successfully
- We set the roots, steps, code, logic for future A/B Testing
- We reached reasonable conclusions about the subject
- We determine that the results are no statistical significant to reject the null hypothesis
- Work with subjects that we don't have control over the databases (FCI shareholders)
- We didn't check if the clients effectively open the emails
- We selected a subject with a very low conversion rate

▼ CX MEETING

AB Testing

- **AB Testing** is the statistical methodology used to determine whether the difference between **variants** is **statistically significant!**
- **Example:** You run a website and want to change the color of the banner on the homepage.
 - **Control group users** will see the existing colour
 - **Variant group users** will see a new colour
 - When the test is over, you can see **if the conversion rate has increased**



▼ SAMPLEO

CON R

```
# Perform inner-join
no_promo <- inner_join(a, b, by = "email")

balanceo <- dbGetQuery(db_handle, "
select
cliente,
COALESCE(email,email_corregido_enriquecido) as email,
segmento_planeam,
Segmento_riesgos,
cantidad_productos,
cliente_activo,
FechaUltMovCA,
SaldoPromedioCAPesos,
SaldoPromedioCADolares,
edad,
sexo,
CategoriaProducto,
Empresa,
--Es_monoproducto,
CantidadPresPers,
CantidadPresHipotec,
CantidadPresEmpresa,
CantidadPresPersUVAS,
CantidadPresRetHab,
CantidadPresPrend,
--Detalle_prestamos,
CantCAPesos,
CantCADolares,
CantCtaCtePesos,
--Detalle_cuentas,
CantCajadeSeguridad,
SegATM,
SegAuto,
SegVida,
SegVivienda,
--Detalle_Seguros,
```

```

TCVisa,
TCMaster
--Detalle_TC,
--antiguedad_anios
from datamining.dbo.wh_clientes_ultima_enriquecida_1344
where email <> ''
")

no_promo_balanceado <- merge(no_promo, balanceo, by.x = 'email',
                               by.y = 'email')

set.seed(1000)
sample <- sample.int( n = nrow(no_promo_balanceado), size = 1)

sample1 <- no_promo_balanceado[sample, ]
sample2 <- no_promo_balanceado[-sample, ]

summary(sample1)
summary(sample2)

```

CON PYTHON

```

# Execute SQL query and fetch data into a DataFrame
balanceo = """
SELECT
    cliente,
    COALESCE(email,email_corregido_enriquecido) as email,
    segmento_planeam,
    Segmento_riesgos,
    cantidad_productos,
    cliente_activo,
    FechaUltMovCA,
    SaldoPromedioCAPesos,
    SaldoPromedioCADolares,
    edad,
    sexo,
    CategoriaProducto,
    Empresa,

```

```

CantidadPresPers,
CantidadPresHipotec,
CantidadPresEmpresa,
CantidadPresPersUVAs,
CantidadPresRetHab,
CantidadPresPrend,
CantCAPesos,
CantCADolares,
CantCtaCtePesos,
CantCajaDeSeguridad,
SegATM,
SegAuto,
SegVida,
SegVivienda,
TCVisa,
TCMaster
FROM datamining.dbo.wh_clientes_ultima_enriquecida_1344
WHERE email <> ''
"""

# Merge DataFrames on 'email' column
no_promo_balanceado = pd.merge(no_promo, balanceo, on='email')

# Set seed for reproducibility
np.random.seed(1000)

# Create a random sample index
sample = np.random.choice(no_promo_balanceado.index, size=int(0.5 * len(no_promo_balanceado)))

# Split the DataFrame into two samples
sample1 = no_promo_balanceado.loc[sample]
sample2 = no_promo_balanceado.loc[~no_promo_balanceado.index.isin(sample)]

# Display summaries
print(sample1.describe())
print(sample2.describe())

```

▼ TESTEO

<https://docs.google.com/document/d/1H4wEVBqjyVFFMIDqig9jBO93GdopR1Wuv6eGd-nwO9w/edit?usp=sharing>

▼ **Brief resultados**

Recolección resultados

▼ **T&L - NEW PROJECT WITH A PARTNER**

Automatización de subida de data.

Primero se debe tener completa la base de diccionario de cruce, sino pueden quedarme clientes sin enmascarar. Entonces, el primer paso es automatizar ese proceso de generación de diccionario de enmascaramiento.

Este proceso se corre todos los 15, para tener margen de error de fecha de actualizacion de la WH que es el lunes siguiente al 5to dia habil

Una vez actualizado eso, puedo automatizar la carga de WH SOCIODEMOGRAFICO Y tarjetas.

En el caso de la WH debo tomar el mes corriente y se debe correr el 15 del mes

En el caso de datos de Tarjeta cuando debería correrlo??? también, el 15.

¿Qué períodos cubiertos se deberían utilizar para este análisis?

Debido a que su evento ha agregado fechas, se recomienda que el período de interrupción (el tiempo que hay entre su línea base y el período de análisis) sea mayor que un **semana calendario** para evitar que las Clientes reciban el evento en el período de línea base.

■ Período de línea base (Preperiodo)

Dura 365 días y finaliza 7 días antes del user_onboarding_date [Editar](#)

■ Período de análisis (Posperiodo)

El período de análisis comenzará 7 días después del user_onboarding_date [Editar](#) El período de análisis finalizará 90 días después del user_onboarding_date [Editar](#)

[Usar diferentes períodos cubiertos por conjunto](#)



[Ver períodos cubiertos calculados](#)

DATA REQUEST - Actualización

Tabla	Cuando corre Script	Actualización de tabla	Notificación de error
Sociodemo	el 15 del mes	Primer lunes luego de 5to día hábil	
Trx	1º mes corriente, → 15/21 mes anterior. 8º mes corriente, → 22/31 mes anterior. 15º mes corriente, → 1/7 mes anterior. 22º mes corriente, → 8/14 mes anterior.		
Tarjeta			

▼ Inspiration Dashboard

AB Testing with Power BI - Ben's Blog

A step by step guide with all the DAX formulas and example to implement AB Testing with Power BI.

 <https://datakuity.com/2020/09/29/ab-testing-with-power-bi/>

