

# Package ‘mediaPlanR’

November 19, 2024

**Type** Package

**Version** 0.1.1

**Date** 2024-10-25

**Title** Herramientas para la Planificación de Medios Publicitarios

**Description** Indicadores, modelos y aplicaciones para la planificación de medios.  
Proporciona interfaces interactivas para la  
exploración y análisis de datos de medios publicitarios.

**Manual** <<https://github.com/majesus/mediaPlanR>>

**Author** Manuel J. Sánchez-Franco [aut, cre] [<<https://orcid.org/0000-0002-8042-3550>>]

**Maintainer** Manuel J. Sánchez-Franco <majesus@us.es>

**Repository** GitHub

**URL** <<https://github.com/majesus/mediaPlanR>>

**BugReports** <https://github.com/majesus/mediaPlanR/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Imports** stats,  
extraDistr,  
shiny (>= 1.7.0),  
bslib (>= 0.4.0),  
ggplot2 (>= 3.4.0),  
dplyr (>= 1.0.0),  
readxl,  
readr,  
DT,  
janitor,  
ggrepel,  
factoextra,  
microbenchmark,  
Deriv,  
doBy,  
scales,  
tidyr,  
tibble,

purrr,  
magrittr,  
plotly

**Suggests** knitr,  
rmarkdown,  
testthat (>= 3.0.0),  
devtools

**Config/testthat/edition** 3

**Depends** R (>= 2.10)

**LazyData** true

## Contents

calcular_metricas_medios . . . . .	2
calc_beta_binomial . . . . .	4
calc_binomial . . . . .	6
calc_cpm . . . . .	8
calc_grps . . . . .	9
calc_hofmans . . . . .	11
calc_MBBD . . . . .	13
calc_metheringham . . . . .	15
calc_R1_R2 . . . . .	16
calc_sainsbury . . . . .	18
canonical_expansion_model . . . . .	19
optimizar_d . . . . .	23
optimizar_dc . . . . .	25
optimize_media_plan . . . . .	28
optimize_media_sb . . . . .	30
plot_grp_metricas . . . . .	33
run_aud_util_explorer . . . . .	34
run_beta_binomial_explorer . . . . .	35
run_reach_converg_explorer . . . . .	35

**Index** 37

---

calcular\_metricas\_medios

*Cálculo de métricas de soportes para un plan publicitario*

---

## Description

Calcula métricas de un plan de medios aceptando tanto datos desde CSV como vectores directamente, permitiendo sobrescribir valores específicos del CSV con vectores. Las métricas calculadas incluyen:

1. RP: Alcance ponderado por inserciones y población
2. SOV: Share of Voice como % del total de GRP
3. CPM: Coste por mil contactos
4. C/RP: Coste por punto de RP

5. Audiencia Útil: Audiencia ajustada por índice de utilidad
6. Coste por Contacto Útil: Tarifa dividida entre la audiencia útil

### Usage

```
calcular_metricas_medios(  
  soportes = NULL,  
  audiencias = NULL,  
  tarifas = NULL,  
  ind_utilidad = NULL,  
  inserciones = NULL,  
  pob_total,  
  file = NULL,  
  sep = ",",  
)
```

### Arguments

soportes	Vector de nombres o nombre de la columna en CSV
audiencias	Vector numérico o nombre de la columna en CSV
tarifas	Vector numérico o nombre de la columna en CSV
ind_utilidad	Vector numérico o nombre de la columna en CSV
inserciones	Vector numérico o nombre de la columna en CSV (opcional)
pob_total	Tamaño de la población objetivo
file	Ruta al archivo CSV (opcional)
sep	Separador usado en el CSV (default: ",")

### Value

Un data.frame con las siguientes columnas:

- Soporte: Nombre del medio
- Audiencia\_miles: Audiencia en miles de lectores
- Numero\_Inserciones: Número de inserciones
- RP: Rating Points
- SOV: Share of Voice
- Tarifa\_Pag\_Color: Tarifa página color
- CPM: Coste por mil
- C\_RP: Coste por Rating Point
- Ind\_Utilidad: Índice de utilidad
- Audiencia\_Util\_miles: Audiencia útil en miles
- Coste\_Contacto\_Util: Coste por contacto útil

### See Also

[calc\\_cpm](#) para cálculo de costes por mil (CPM) [calc\\_grps](#) para cálculo de GRPs

## Examples

```
# Ejemplo 1: Usando solo vectores
resultado <- calcular_metricas_medios(
  soportes = c("El País", "El Mundo"),
  audiencias = c(1520000, 780000),
  tarifas = c(39800, 35600),
  ind_utilidad = c(1.2, 1.1),
  pob_total = 39500000
)

## Not run:
# Ejemplo 2: Usando CSV con nombres de columnas por defecto
resultado <- calcular_metricas_medios(
  file = "data.csv",
  soportes = "soportes",
  audiencias = "audiencias",
  tarifas = "tarifas",
  ind_utilidad = "ind_utilidad",
  pob_total = 39500000
)

# Ejemplo 3: Combinando CSV con vector de inserciones personalizado
resultado <- calcular_metricas_medios(
  file = "data.csv",
  soportes = "soportes",
  audiencias = "audiencias",
  tarifas = "tarifas",
  ind_utilidad = "ind_utilidad",
  inserciones = c(2, 3, 1, 2), # Sobrescribe las inserciones del CSV
  pob_total = 39500000
)

## End(Not run)
```

---

calc_beta_binomial	<i>Cálculo de la cobertura y distribución de contactos (y acumulada) usando modelo Beta-Binomial</i>
--------------------	--

---

## Description

Implementa el modelo Beta-Binomial para calcular la audiencia neta acumulada y la distribución de contactos (y acumulada). El modelo Beta-Binomial considera la heterogeneidad en la probabilidad de exposición de los individuos. Combina dos pasos: modela la probabilidad de éxito aplicando la distribución Beta de parámetros alpha y beta -lo cual reduce a dos los datos necesarios para su estimación; y emplea la probabilidad en la distribución Binomial (combinada con la distribución Beta) para valorar la distribución de contactos (y acumulada). Es útil cuando la probabilidad de éxito no es conocida a priori, y puede variar entre los individuos. Los parámetros alpha y beta precisamente permiten ajustar la forma de la distribución para que refleje la incertidumbre en relación con la probabilidad de éxito.

## Usage

```
calc_beta_binomial(A1, A2, P, n)
```

**Arguments**

A1	Audiencia del soporte tras la primera inserción
A2	Audiencia del soporte tras la segunda inserción
P	Tamaño total de la población
n	Número total de inserciones planificadas (debe ser entero positivo)

**Details**

El modelo Beta-Binomial:

1. Calcula los parámetros alpha y beta a partir de A1 y A2
2. Modela la heterogeneidad en la exposición mediante la distribución Beta
3. Combina la distribución Beta con la Binomial para la distribución de contactos
4. Calcula probabilidades exactas para cada nivel de exposición

El proceso incluye:

- Estimación de coeficientes de duplicación R1 y R2
- Cálculo de parámetros alpha y beta del modelo
- Generación de distribución de contactos
- Cálculo de la distribución de contactos (y acumuladas)

**Value**

Una lista "reach\_beta\_binomial" conteniendo:

- reach: Lista con la cobertura:
  - porcentaje: Cobertura en porcentaje
  - personas: Cobertura en número de personas
- distribucion: Lista con la distribución de contactos:
  - porcentaje: Vector con probabilidad de cada número de exposiciones
  - personas: Vector con número de personas para cada número de exposiciones
- acumulada: Lista con la distribución acumulada:
  - porcentaje: Vector con probabilidades acumuladas
  - personas: Vector con número de personas acumuladas al menos i veces
- parametros: Lista con parámetros del modelo:
  - alpha: Parámetro alpha estimado
  - beta: Parámetro beta estimado
  - prob\_cero\_contactos: Probabilidad de no exposición

**Note**

El modelo Beta-Binomial es especialmente adecuado cuando:

- Existe heterogeneidad significativa en la población
- Se dispone de datos de audiencias acumuladas (A1 y A2)

## References

Aldás Manzano, J. (1998). Modelos de determinación de la cobertura y la distribución de contactos en la planificación de medios publicitarios impresos. Tesis doctoral, Universidad de Valencia, España.

## See Also

[calc\\_sainsbury](#) para estimaciones con la distribución Binomial [calc\\_binomial](#) para estimaciones con la distribución Beta-Binomial [calc\\_metheringham](#) para estimaciones con la distribución de Metheringham [calc\\_hofmans](#) para estimaciones con la distribución de Hofmans

## Examples

```
# Ejemplo básico
resultado <- calc_beta_binomial(
  A1 = 500000, # Primera audiencia
  A2 = 550000, # Segunda audiencia
  P = 1000000, # Población total
  n = 5        # Número de inserciones
)

# Examinar resultados
print(paste("Cobertura:", round(resultado$reach$porcentaje, 2), "%"))
print(paste("Alpha:", round(resultado$parametros$alpha, 4)))
print(paste("Beta:", round(resultado$parametros$beta, 4)))

# Verificar consistencia de las distribuciones
## Not run:
sum_dist <- sum(resultado$distribucion$porcentaje)/100
print(paste("Suma distribución:", round(sum_dist +
  resultado$parametros$prob_cero_contactos/100, 4)))

## End(Not run)
```

---

calc_binomial	<i>Cálculo de cobertura y distribución de contactos (y acumulada) según modelo Binomial</i>
---------------	---

---

## Description

Implementa el modelo Binomial, desarrollado por Chandon (1985), para calcular la cobertura y distribución de contactos (y acumulada) de plan de medios de  $n$  soportes y una única inserción por soporte. El modelo Binomial asume la duplicación aleatoria (i.e., la exposición a un soporte no modifica la probabilidad de resultar expuesto a otro), y la homogeneidad de las probabilidades de exposición del soporte y las probabilidades individuales de exposición. Uniendo estas dos hipótesis últimas, la probabilidad de exposición de cualquier individuo a un soporte determinado se calcula como la media de las audiencias de cada soporte. Las probabilidades de exposición son estacionarias respecto al tiempo.

## Usage

```
calc_binomial(audiencias, pob_total)
```

## Arguments

audiencias	Vector numérico con las audiencias individuales de cada soporte
pob_total	Tamaño de la población

## Details

El modelo Bnomial calcula:

1. Cobertura considerando un soporte hipotético “promedio” cuya audiencia es la media simple de las audiencias de cada soporte
2. Distribución de contactos para cada nivel de exposición
3. Distribución de contactos acumulada (expuestos al menos  $i$  veces)

La metodología incluye:

- Conversión de audiencias a probabilidades individuales
- Cálculo de probabilidad media de exposición
- Aplicación del modelo Binomial para  $n$  inserciones
- Cálculo de distribuciones de contactos (y acumulada)

## Value

Una lista "reach\_binomial" conteniendo:

- reach: Lista con la cobertura:
  - porcentaje: Cobertura en porcentaje
  - personas: Cobertura en número de personas
- distribucion: Lista con la distribución de contactos:
  - porcentaje: Vector con probabilidad de cada número de exposiciones
  - personas: Vector con número de personas para cada número de exposiciones
- acumulada: Lista con la distribución acumulada:
  - porcentaje: Vector con probabilidades acumuladas
  - personas: Vector con número de personas acumuladas al menos  $i$  veces

## References

Aldás Manzano, J. (1998). Modelos de determinación de la cobertura y la distribución de contactos en la planificación de medios publicitarios impresos. Tesis doctoral, Universidad de Valencia, España.

## See Also

[calc\\_sainsbury](#) para estimaciones con la distribución Binomial [calc\\_beta\\_binomial](#) para estimaciones con la distribución Beta-Binomial [calc\\_metheringham](#) para estimaciones con la distribución de Metheringham [calc\\_hofmans](#) para estimaciones con la distribución de Hofmans

## Examples

```
# Ejemplo básico con tres soportes
audiencias <- c(300000, 400000, 200000)
pob_total <- 1000000
resultado <- calc_binomial(audiencias, pob_total)

# Examinar los resultados
print(paste("Cobertura total:", resultado$reach$porcentaje, "%"))
print(paste("Probabilidad media:", resultado$probabilidad_media))

# Verificar que las distribuciones suman 1 (100%)
## Not run:
sum_dist <- sum(resultado$distribucion$porcentaje)/100
print(paste("Suma distribución:", round(sum_dist, 4)))

## End(Not run)
```

---

calc\_cpm

Cálculo de CPM para plan de medios o soportes individuales

---

## Description

Calcula el Coste Por Mil (CPM) ya sea para un plan de medios completo o para soportes individuales, permitiendo evaluar la eficiencia en términos de coste por cada mil personas alcanzadas, es decir, permite comparar la rentabilidad de diferentes estrategias y medios dentro del mismo plan de campaña.

## Usage

```
calc_cpm(precios, audiencias = NULL, cobertura = NULL, tipo = "soporte")
```

## Arguments

precios	Vector numérico con precios de cada inserción o precio total (presupuesto) del plan
audiencias	Vector numérico con audiencias de cada soporte
cobertura	Opcional. Cobertura del plan en personas
tipo	Character. Tipo de cálculo: "soporte" o "plan" (default: "soporte")

## Details

El CPM se puede calcular de dos formas:

1. Para soportes individuales:
  - $CPM = (\text{Precio inserción} / \text{Audiencia}) \times 1000$
2. Para plan completo:
  - $CPM = (\text{Precio total} / \text{Cobertura}) \times 1000$



**Value**

Una lista conteniendo:

- cpm: Vector de CPMs calculados o CPM del plan
- tipo: Tipo de cálculo realizado
- total: Suma total de precios (si aplica)

**Note**

El CPM es útil para:

- Comparar eficiencia entre soportes
- Evaluar rentabilidad de planes de medios
- Optimizar presupuestos publicitarios

**See Also**

[calc\\_grps](#) para cálculo de GRPs

**Examples**

```
# CPM por soportes
cpm1 <- calc_cpm(
  precios = c(1000, 1500, 800),
  audiencias = c(300000, 400000, 200000)
)

# CPM del plan completo
cpm2 <- calc_cpm(
  precios = 25000,
  cobertura = 750000,
  tipo = "plan"
)
```

---

calc\_grps

*Cálculo de GRPs mediante la cobertura y frecuencia media, o el cálculo de las impresiones totales*

---

**Description**

Calcula los Gross Rating Points (GRPs) de un plan de medios utilizando dos métodos diferentes: mediante impresiones totales o mediante cobertura y frecuencia media. Los GRP (Gross Rating Points) son una métrica publicitaria que indica el impacto total de una campaña sobre una audiencia determinada, expresando la suma del alcance por la frecuencia de exposición. Se calculan dividiendo el número total de impresiones (contactos o veces que el anuncio fue visto) por la población relevante, multiplicado por 100, lo cual permite expresar la exposición acumulativa de la campaña como un porcentaje.

**Usage**

```
calc_grps(
  audiencias,
  inserciones,
  pob_total,
  cobertura = NULL,
  metodo = "impresiones"
)
```

**Arguments**

audiencias	Vector numérico con las audiencias de cada soporte
inserciones	Vector numérico del número de inserciones por soporte
pob_total	Tamaño de la población
cobertura	Opcional. Cobertura en porcentaje (si se conoce)
metodo	Character. Método de cálculo: "impresiones" o "cobertura" (default: "impresiones")

**Details**

El cálculo se puede realizar mediante dos métodos:

1. Método por impresiones:
  - Calcula impresiones totales:  $\text{SUMATORIO}(\text{Audiencia}_i \times \text{Inserciones}_i)$
  - $\text{GRPs} = (\text{Impresiones} / \text{Población}) \times 100$
2. Método por cobertura:
  - $\text{Frecuencia media} = \text{Impresiones totales} / (\text{Cobertura} \times \text{Población})$
  - $\text{GRPs} = \text{Cobertura} \times \text{Frecuencia media}$

**Value**

Una lista conteniendo:

- grps: Valor de GRPs calculado
- impresiones\_totales: Suma total de impresiones
- frecuencia\_media: Frecuencia media (si aplica)
- metodo: Método utilizado para el cálculo

**Note**

Los GRPs son una medida de presión publicitaria que:

- Pueden superar el 100%
- Indican el número de impactos por cada 100 personas de la población
- Son útiles para comparar campañas de publicidad
- Su debilidad reside en que campañas con diferentes valores de cobertura % y frecuencia media pueden arrojar un mismo nivel de GRPs

**See Also**

[calc\\_cpm](#) para cálculo de costes por mil (CPM)

## Examples

```
# Cálculo por método de impresiones
grps1 <- calc_grps(
  audiencias = c(300000, 400000, 200000),
  inserciones = c(3, 2, 4),
  pob_total = 1000000
)

# Cálculo por método de cobertura
grps2 <- calc_grps(
  audiencias = c(300000, 400000, 200000),
  inserciones = c(3, 2, 4),
  pob_total = 1000000,
  cobertura = 65.5,
  metodo = "cobertura"
)
```

---

calc\_hofmans

*Cálculo de audiencia acumulada según el modelo de audiencia acumulada de Hofmans*

---

## Description

Implementa el modelo de Hofmans (1966) para calcular la audiencia acumulada de un plan de medios con múltiples inserciones en un soporte. El modelo considera la duplicación entre inserciones, y utiliza un parámetro de ajuste (alpha) para mejorar la estimación de las audiencias acumuladas.

## Usage

```
calc_hofmans(R1, R2, N, show_steps = TRUE)
```

## Arguments

R1	Numérico. Cobertura tras la primera inserción (como proporción entre 0 y 1)
R2	Numérico. Cobertura tras la segunda inserción (como proporción entre 0 y 1)
N	Entero. Número de inserciones para las que calcular la audiencia acumulada
show_steps	Lógico. Si TRUE muestra los pasos intermedios del cálculo

## Details

El modelo de Hofmans calcula la cobertura acumulada en dos etapas:

1. Utiliza una primera formulación para calcular R3:
  - $R3 = (3R1)^2 / (3R1 + k(2R1-R2)(3 \text{ choose } 2))$
  - donde  $k = 2R1/R2$
2. Para  $N > 3$  aplica una formulación mejorada que incorpora un parámetro alpha:
  - $RN = (NR1)^2 / (NR1 + k*(N-1)^a*(N/2)*d)$
  - donde alpha se calcula usando R3

- y  $d = 2R1 - R2$  es la duplicación entre inserciones

El modelo asume:

- Audiencia constante para todas las inserciones
- Duplicación constante entre pares de inserciones
- Comportamiento no lineal de la acumulación para  $N > 3$

## Value

Una lista "hofmans\_reach" conteniendo:

- resultados: Data frame con:
  - N: Número de inserción
  - RN: Cobertura acumulada (proporción)
- parametros: Lista con los parámetros calculados:
  - k: Factor k calculado
  - d: Duplicación entre inserciones
  - alpha: Parámetro de ajuste para  $N > 3$
- plot: Gráfico de la evolución de la cobertura

## References

Aldás Manzano, J. (1998). Modelos de determinación de la cobertura y la distribución de contactos en la planificación de medios publicitarios impresos. Tesis doctoral, Universidad de Valencia, España.

## See Also

[calc\\_beta\\_binomial](#) para estimaciones con la distribución Beta-Binomial [calc\\_sainsbury](#) para estimaciones el modelo de Sainsbury [calc\\_binomial](#) para estimaciones con el modelo Binomial [calc\\_metheringham](#) para estimaciones con el modelo de Metheringham

## Examples

```
# Ejemplo básico con 5 inserciones
R1 <- 0.06    # 6% de cobertura primera inserción
R2 <- 0.103   # 10.3% de cobertura segunda inserción
resultado <- calc_hofmans(R1, R2, N = 5)

# Examinar los resultados
print(resultado$resultados)
print(resultado$parametros)

# Ejemplo con validación de datos
## Not run:
R1_invalido <- 1.2 # >100% cobertura
resultado <- calc_hofmans(R1_invalido, R2, N = 5)
# Generará un error por cobertura inválida

## End(Not run)
```

calc\_MBBD

*Cálculo del Modelo Morgensztem Beta Binomial Distribution (MBBD)***Description**

Implementa el modelo MBBB para calcular la distribución de contactos de un plan de medios. Combina la estimación de cobertura de Morgensztem con la distribución beta binomial para ajustar la distribución de contactos.

**Usage**

```
calc_MBBD(
  m,
  insertions,
  audiences,
  RM,
  universe,
  A0,
  precision = 100,
  max_iter = 100,
  adj_factor = 0.01
)
```

**Arguments**

m	Integer. Número de soportes
insertions	Vector numérico. Número de inserciones para cada soporte (ni)
audiences	Vector numérico. Audiencia de cada soporte en personas (Ai)
RM	Entero. Estimación de cobertura según Morgensztem en personas
universe	Entero. Tamaño del universo objetivo en personas
A0	Numérico. Valor inicial del parámetro A (entre 0 y 10)
precision	Numérico. Criterio de convergencia en personas. Por defecto 100
max_iter	Entero. Número máximo de iteraciones permitidas. Por defecto 100
adj_factor	Numérico. Factor de ajuste para el parámetro A. Por defecto 0.01

**Details**

El modelo MBBB ajusta iterativamente los parámetros de una distribución beta binomial hasta que su cobertura coincide con la estimada por el método de Morgensztem:

1. Calcula B0 inicial según la fórmula:
  - $B0 = A0 * (SUMATORIO ni - SUMATORIO niAi) / (SUMATORIO niAi)$
2. Ajusta iterativamente los parámetros hasta que las coberturas convergen:
  - Si RM mayor que BBD: aumenta A
  - Si RM menor que BBD: disminuye A
  - Recalcula B en cada iteración

El modelo asume:

- Los parámetros A y B deben estar entre 0 y 10
- La cobertura BBD se calcula como  $1 - P(K=0)$
- La convergencia se alcanza cuando  $|BBD - RM|$  menor o igual que precision

## Value

Una lista de clase "MBBD" conteniendo:

- parameters: Lista con parámetros finales:
  - AF: Parámetro A final
  - BF: Parámetro B final
  - N: Total de inserciones
  - universe: Tamaño del universo
  - iterations: Número de iteraciones realizadas
  - converged: Indicador de convergencia
- coverage: Lista con coberturas:
  - RM: Cobertura de Morgensztem
  - BBD: Cobertura Beta Binomial
- contact\_distribution: Vector con probabilidades de 0 a N contactos
- iteration\_history: Data frame con historial de iteraciones

## References

Aldás Manzano, J. (1998). Modelos de determinación de la cobertura y la distribución de contactos en la planificación de medios publicitarios impresos. Tesis doctoral, Universidad de Valencia, España.

## See Also

[calc\\_beta\\_binomial](#) para estimaciones con la distribución Beta-Binomial [calc\\_sainsbury](#) para estimaciones el modelo de Sainsbury [calc\\_binomial](#) para estimaciones con el modelo Binomial [calc\\_metheringham](#) para estimaciones con el modelo de Metheringham

## Examples

```
Ejemplo básico
m <- 3
insertions <- c(5, 7, 4)
audiences <- c(500000, 550000, 600000)
RM <- 550000
universe <- 1000000 # Añadimos el parámetro universe que faltaba
resultado <- calc_MBBD(m, insertions, audiencias, RM, universe, A0=0.1)

# Examinar resultados
print(resultado)
```

## Description

Calcula métricas fundamentales para la aplicación del modelo de Metheringham, incluyendo la audiencia media (A1), duplicación media (D) y audiencia tras la segunda exposición en el hipotético soporte promedio (A2). El modelo de Metheringham (1964) se basa en que los individuos tienen probabilidades heterogéneas que se distribuyen como una distribución Beta para el conjunto. Los soportes son homogéneos (a saber, todos los soportes acaban con la misma distribución Beta de probabilidades de exposición). La acumulación y duplicación de audiencias se promedian entre los soportes para diseñar un soporte hipotético promedio.

## Usage

```
calc_metheringham(audiencias, inserciones, vec_duplicacion, ayuda = TRUE)
```

## Arguments

audiencias	Vector numérico con las audiencias de cada soporte
inserciones	Vector numérico con el número de inserciones por soporte
vec_duplicacion	Vector numérico con valores de duplicación entre soportes
ayuda	Logical. Si TRUE, muestra una guía de uso detallada (default: TRUE)

## Details

La función realiza los siguientes cálculos principales:

1. Audiencia media tras la primera inserción (A1):
  - Calcula la media ponderada de audiencias por número de inserciones
  - Fórmula:  $A1 = \text{SUMATORIO}(\text{Audiencia}_i \times \text{Inserciones}_i) / \text{SUMATORIO}(\text{Inserciones}_i)$
2. Duplicación media (D):
  - Calcula la media ponderada de duplicaciones por oportunidades de contacto
  - Considera todas las combinaciones posibles entre soportes ii, ij
3. Audiencia tras la segunda inserción (A2):
  - Calcula la audiencia que se expone al menos una vez tras la segunda inserción
  - Fórmula:  $A2 = 2 \times A1 - D$

## Value

Una lista conteniendo:

- `audiencia_media`: Media ponderada de audiencias (A1)
- `duplicacion_media`: Media ponderada de duplicaciones (D)
- `audiencia_segunda`: Audiencia tras la segunda inserción (A2)
- `vector_oportunidades`: Vector que contiene el número de oportunidades de contacto entre pares de inserciones, siguiendo el mismo orden que el `vec_duplicacion`

**Note**

El vector de duplicación debe seguir un orden específico:

- Para  $n$  soportes, se requieren  $n*(n+1)/2$  valores
- Los valores se ordenan por filas de la matriz triangular superior
- Incluye la duplicación de cada soporte consigo mismo
- El orden sigue el patrón: (1,1), (1,2), (1,3), (2,2), (2,3), (3,3)

**References**

Aldás Manzano, J. (1998). Modelos de determinación de la cobertura y la distribución de contactos en la planificación de medios publicitarios impresos. Tesis doctoral, Universidad de Valencia, España.

**See Also**

[calc\\_sainsbury](#) para estimaciones con la distribución Binomial [calc\\_binomial](#) para estimaciones con la distribución Beta-Binomial [calc\\_beta\\_binomial](#) para estimaciones con la distribución de Metherringham [calc\\_hofmans](#) para estimaciones con la distribución de Hofmans

**Examples**

```
# Ejemplo básico con tres soportes
metricas <- calc_metherringham(
  audiencias = c(1500000, 800000, 1200000),
  inserciones = c(4, 3, 5),
  vec_duplicacion = c(150000, 200000, 180000,
                     120000, 140000,
                     170000),
  ayuda = FALSE
)

# Mostrar solo la guía de uso
calc_metherringham(
  audiencias = NULL,
  inserciones = NULL,
  vec_duplicacion = NULL,
  ayuda = TRUE
)
```

---

calc\_R1\_R2

---

Cálculo de los valores R1 y R2 (modelo: Beta-Binomial)

---

**Description**

Calcula los valores R1 y R2 a partir de los parámetros de forma alpha y beta del modelo de audiencia neta acumulada Beta-Binomial. Los valores son clave para evaluar la audiencia neta y la distribución de contactos (y acumulada). Si la probabilidad de éxito se distribuye según una distribución beta de parámetros alpha y beta, la distribución de contactos, es una distribución compuesta: la distribución beta binomial.



**Usage**

```
calc_R1_R2(A, B)
```

**Arguments**

A	Parámetro de forma alpha, debe ser numérico y positivo
B	Parámetro de forma beta, debe ser numérico y positivo

**Details**

Los coeficientes R1 y R2 son medidas de la duplicación de audiencias:

- R1 mide el tanto por uno de personas alcanzadas tras la primera inserción en el soporte elegido
- R2 mide el tanto por uno de personas alcanzadas tras la segunda inserción en el soporte elegido

El proceso de cálculo:

1. Calcula R1 directamente como  $A/(A+B)$
2. Optimiza R2 mediante un proceso iterativo
3. Verifica que los valores R1 y R2 estén en el rango (0,1)

**Value**

Una lista con dos componentes:

- R1: Coeficiente (tanto por uno) de audiencia acumulada tras la primera inserción
- R2: Coeficiente (tanto por uno) de audiencia acumulada tras la segunda inserción

**See Also**

[calc\\_beta\\_binomial](#) para estimaciones con el modelo Binomial [calc\\_sainsbury](#) para estimaciones con el modelo de Sainsbury

**Examples**

```
# Calcular R1 y R2 para alpha=0.5 y beta=0.3
resultados <- calc_R1_R2(0.5, 0.3)

# Ver resultados
print(paste("R1:", round(resultados$R1, 4)))
print(paste("R2:", round(resultados$R2, 4)))

# Verificar que los valores están en el rango esperado
stopifnot(resultados$R1 >= 0, resultados$R1 <= 1)
stopifnot(resultados$R2 >= 0, resultados$R2 <= 1)
```

---

calc_sainsbury	<i>Cálculo de cobertura y distribución de contactos (y acumulada) según modelo de Sainsbury</i>
----------------	---

---

## Description

Implementa el modelo de Sainsbury, desarrollado por E. J. Sansbury en la London Press Exchange, para calcular la cobertura y la distribución de contactos para un conjunto de soportes publicitarios y una única inserción por soporte. El modelo considera la duplicación aleatoria, las probabilidades individuales de exposición homogéneas, y las probabilidades de exposición del soporte heterogéneas para una estimación más precisa de la cobertura y la distribución de contactos (y acumulada). De las dos últimas hipótesis se deriva que la probabilidad de que un individuo resulte expuesto al soporte  $i$  vendrá dado por el cociente entre la audiencia del soporte  $i$  (casos favorables) y la población (casos totales). Por su parte, de la asunción de la duplicación aleatoria se deriva que la probabilidad de exposición continuará siendo una variable Bernoulli con diferentes probabilidades de exposición en cada soporte.

## Usage

```
calc_sainsbury(audiencias, pob_total)
```

## Arguments

audiencias	Vector numérico con las audiencias individuales de cada soporte
pob_total	Tamaño de la población

## Details

El modelo de Sainsbury simplificado calcula:

1. Cobertura considerando la duplicación entre soportes como el producto de las probabilidades individuales
2. Distribución de contactos para cada nivel de exposición  $i$
3. Distribución de contactos acumulada (expuestos al menos  $i$  veces)

El proceso incluye:

- Conversión de audiencias a probabilidades
- Cálculo de las posibles combinaciones de soportes
- Estimación de probabilidades conjuntas
- Agregación de resultados: distribución de contactos (y acumulada)

## Value

Una lista "reach\_sainsbury" conteniendo:

- reach: Lista con la cobertura:
  - porcentaje: Cobertura en porcentaje
  - personas: Cobertura en número de personas
- distribucion: Lista con la distribución de contactos:

- porcentaje: Vector con probabilidad de cada número de exposiciones
- personas: Vector con número de personas para cada número de exposiciones
- acumulada: Lista con la distribución acumulada:
  - porcentaje: Vector con probabilidades acumuladas
  - personas: Vector con número de personas acumuladas al menos i veces

## References

Aldás Manzano, J. (1998). Modelos de determinación de la cobertura y la distribución de contactos en la planificación de medios publicitarios impresos. Tesis doctoral, Universidad de Valencia, España.

## See Also

[calc\\_binomial](#) para estimaciones con la distribución Binomial [calc\\_beta\\_binomial](#) para estimaciones con la distribución Beta-Binomial [calc\\_metheringham](#) para estimaciones con la distribución de Metheringham [calc\\_hofmans](#) para estimaciones con la distribución de Hofmans

## Examples

```
# Ejemplo básico con tres soportes
audiencias <- c(300000, 400000, 200000)
pob_total <- 1000000
resultado <- calc_sainsbury(audiencias, pob_total)

# Examinar los resultados
print(resultado$reach$porcentaje) # Cobertura en porcentaje
print(resultado$distribucion$personas) # Personas por número de contactos

# Ejemplo con validación de datos
## Not run:
audiencias_invalidas <- c(300000, -400000, 200000)
resultado <- calc_sainsbury(audiencias_invalidas, pob_total)
# Generará un error por audiencia negativa

## End(Not run)
```

---

canonical\_expansion\_model

*Modelo de Expansión Canónica para Distribuciones de Exposición a Medios*

---

## Description

Implementa el modelo de expansión canónica desarrollado por Danaher (1991) para calcular la distribución de contactos en planes de medios con múltiples inserciones por soporte. El modelo considera las correlaciones entre soportes y permite estimar la cobertura y distribución de contactos de manera más precisa que los modelos tradicionales al incorporar las dependencias entre exposiciones.

## Usage

```
canonical_expansion_model(  
    marginal_probs,  
    correlations,  
    insertions,  
    population = 1e+07,  
    truncation_order = 2,  
    tolerance = 1e-10  
)
```

## Arguments

marginal_probs	Vector numérico con las probabilidades marginales de exposición por vehículo
correlations	Matriz numérica de correlaciones
insertions	Vector numérico con el número de inserciones por vehículo
population	Tamaño de la población objetivo (por defecto 1000000)
truncation_order	Orden de truncación para la expansión canónica (por defecto 2). Controla cuántos términos de la expansión se incluyen en el cálculo.
tolerance	Tolerancia numérica para validaciones
vehicle_data	DataFrame con datos de vehículos

## Details

El modelo de expansión canónica calcula:

1. Distribución conjunta de exposiciones considerando correlaciones entre soportes
2. Distribución de contactos para cada nivel de exposición
3. Métricas agregadas como cobertura total y GRPs

El proceso incluye:

- Validación exhaustiva de inputs:
  - Verificación de dimensiones y consistencia entre inputs
  - Validación de rangos (probabilidades entre 0 y 1)
  - Verificación de matriz de correlaciones (simetría, semidefinida positiva)
  - Control de tipos de datos y valores permitidos
- Cálculo de variables canónicas según ecuaciones 2 y 3 del paper original
- Estimación de medias y varianzas para las distribuciones BBD marginales
- Cálculo de probabilidades conjuntas según ecuación 6
- Agregación de resultados en distribuciones de contactos y métricas

La precisión numérica del modelo se controla mediante el parámetro `tolerance`, que afecta tanto a las validaciones como a los cálculos internos.

**Value**

Una lista de clase "reach\_canonical" conteniendo:

- reach: Lista con la cobertura:
  - porcentaje: Cobertura neta en porcentaje
  - personas: Cobertura neta en número de personas
- distribucion: Lista con la distribución de contactos:
  - porcentaje: Vector con probabilidad de cada número de exposiciones
  - personas: Vector con número de personas para cada número de exposiciones
- acumulada: Lista con la distribución acumulada:
  - porcentaje: Vector con probabilidades acumuladas
  - personas: Vector con personas alcanzadas al menos N veces
- parametros: Lista con parámetros del modelo:
  - probabilidad\_media: Media de las probabilidades marginales
  - correlaciones: Matriz de correlaciones entre soportes
  - inserciones: Vector de inserciones por soporte
  - prob\_cero\_contactos: Probabilidad de no exposición
  - grps: Gross Rating Points totales
  - bbd\_params: Parámetros de las distribuciones Beta-Binomial para cada soporte

Lista con validación y DataFrame de correlaciones únicas Validación y ordenamiento de inputs principales

Lista con datos validados y ordenados Cálculo de parámetros BBD

Lista de parámetros BBD por vehículo Cálculo de matriz canónica

Matriz de variables canónicas normalizadas Cálculo de probabilidad conjunta

Probabilidad conjunta para un patrón de exposición Genera todos los patrones posibles de exposición

**References**

Danaher, P. J. (1991). A canonical expansion model for multivariate media exposure distributions: A generalization of the Duplication of Viewing Law. *Journal of Marketing Research*, 28(3), 361-367.

**See Also**

[calc\\_sainsbury](#) para estimaciones con duplicación aleatoria [calc\\_binomial](#) para estimaciones con la distribución Binomial [calc\\_beta\\_binomial](#) para estimaciones con la distribución Beta-Binomial [calc\\_metheringham](#) para estimaciones con la distribución de Metheringham [calc\\_hofmans](#) para estimaciones con la distribución de Hofmans Valida y extrae correlaciones únicas de una matriz de correlaciones

**Examples**

```
# Ejemplo básico con tres soportes
marginal_probs <- c(0.25, 0.20, 0.15)
correlations <- matrix(
  c(1.000, 0.085, 0.075,
    0.085, 1.000, 0.080,
```

```

    0.075, 0.080, 1.000),
  nrow = 3,
  byrow = TRUE
)
insertions <- c(8, 6, 4)
population <- 1000000

# Ejemplo con parámetros por defecto
resultados <- canonical_expansion_model(
  marginal_probs = marginal_probs,
  correlations = correlations,
  insertions = insertions,
  population = population
)

# Ejemplo con parámetros adicionales de control
resultados_precision <- canonical_expansion_model(
  marginal_probs = marginal_probs,
  correlations = correlations,
  insertions = insertions,
  population = population,
  truncation_order = 2,
  tolerance = 1e-10
)

# Examinar los resultados
print(resultados)

# Ejemplo con validación de datos
## Not run:
# Matriz de correlaciones inválida
correlations_invalidas <- matrix(
  c(1.000, 0.085, 0.075,
    0.085, 1.000),
  nrow = 2,
  byrow = TRUE
)
resultados <- canonical_expansion_model(
  marginal_probs = marginal_probs,
  correlations = correlations_invalidas,
  insertions = insertions
)
# Generará un error por dimensiones incorrectas

# Probabilidades marginales inválidas
probs_invalidas <- c(1.2, 0.2, 0.15)
resultados <- canonical_expansion_model(
  marginal_probs = probs_invalidas,
  correlations = correlations,
  insertions = insertions
)
# Generará un error por probabilidades fuera de rango

## End(Not run)

```

---

optimizar\_d*Optimización de distribución de contactos mediante modelo Beta-Binomial*

---

### Description

Esta función optimiza la distribución de contactos publicitarios y calcula los coeficientes de duplicación (R1 y R2) utilizando la distribución Beta-Binomial. El proceso busca la mejor combinación de parámetros alpha y beta y número de inserciones que satisfaga los criterios de cobertura efectiva y frecuencia efectiva (FE) especificados por el usuario.

### Usage

```
optimizar_d(  
  Pob,  
  FE,  
  cob_efectiva,  
  A1,  
  max_inserciones,  
  tolerancia = 0.05,  
  step_A = 0.1,  
  step_B = 0.1,  
  batch_size = 1e+06,  
  min_soluciones = 10,  
  error_aceptable = 0.01  
)
```

### Arguments

Pob	Tamaño de la población
FE	Frecuencia efectiva (FE, número objetivo de impactos por persona)
cob_efectiva	Número objetivo de personas a alcanzar con FE contactos
A1	Audiencia tras la primera inserción
max_inserciones	Número de inserciones máximo a considerar (default: 5)
tolerancia	Margen de error permitido en las soluciones (default: 0.05)
step_A	Incremento para búsqueda del parámetro alpha (default: 0.025)
step_B	Incremento para búsqueda del parámetro beta (default: 0.025)
batch_size	Tamaño del lote para procesamiento (default: 1000000)
min_soluciones	Número mínimo de soluciones para parar (default: 10)
error_aceptable	Error aceptable como proporción (default: 0.01)

### Details

La función realiza los siguientes pasos:

1. Validación de parámetros de entrada y dependencias

2. Cálculo de valores objetivo normalizados y tolerancias
3. Generación de combinaciones de parámetros (alpha, beta, n)
4. Cálculo de distribuciones Beta-Binomiales por lotes
5. Filtrado de resultados según criterios especificados
6. Cálculo de coeficientes R1 y R2 para soluciones válidas
7. Selección de mejor solución y generación de distribución final

## Value

Una lista con los siguientes componentes:

- mejores\_combinaciones: Data frame con todas las combinaciones válidas de parámetros, incluyendo:
  - n: Número de inserciones
  - x: Frecuencia efectiva
  - alpha: Parámetro alpha del modelo
  - beta: Parámetro beta del modelo
  - R1: Proporción de personas alcanzadas tras la primera inserción
  - R2: Proporción de personas alcanzadas tras la segunda inserción
  - prob: Probabilidad asociada
  - distancia\_objetivo: Diferencia absoluta con respecto al objetivo
- mejores\_combinaciones\_top\_10: Las 10 mejores combinaciones según criterios
- data: Data frame con la distribución de contactos final
- alpha: Valor óptimo seleccionado para alpha
- beta: Valor óptimo seleccionado para beta
- n\_optimo: Número óptimo de inserciones

## Note

Los parámetros alpha y beta controlan la forma de la distribución Beta-Binomial:

- alpha: Controla la asimetría hacia valores altos de probabilidad
- beta: Controla la asimetría hacia valores bajos de probabilidad
- La combinación de ambos determina la dispersión y forma final

La función utiliza procesamiento por lotes para optimizar el uso de memoria y proporciona información de progreso durante la ejecución.

## References

Leckenby, J. D., & Boyd, M. M. (1984). An improved beta binomial reach/frequency model for magazines. *Current Issues and Research in Advertising*, 7(1), 1-24.

## See Also

[calc\\_R1\\_R2](#) para detalles sobre el cálculo de coeficientes de duplicación



## Examples

```
# Ejemplo 1: Caso básico con población pequeña
resultado1 <- optimizar_d(
  Pob = 100000,      # Población de 100 mil
  FE = 3,            # Frecuencia efectiva de 3
  cob_efectiva = 59000, # Objetivo: 59% de cobertura
  A1 = 50000,        # Primera audiencia: 50%
  max_inserciones = 5  # Máximo 5 inserciones
)

# Examinar resultados
print(head(resultado1$mejores_combinaciones))
print(resultado1$data)

# Ejemplo 2: Caso con mayor precisión
resultado2 <- optimizar_d(
  Pob = 1000000,
  FE = 4,
  cob_efectiva = 600000,
  A1 = 450000,
  max_inserciones = 8,
  tolerancia = 0.03,    # Menor tolerancia
  step_A = 0.01,        # Pasos más pequeños
  step_B = 0.01,
  min_soluciones = 20   # Más soluciones requeridas
)
```

---

optimizar\_dc

*Optimización de distribución de contactos acumulada mediante modelo Beta-Binomial*

---

## Description

Esta función optimiza la distribución de contactos publicitarios y calcula los coeficientes de duplicación (R1 y R2) utilizando la distribución Beta-Binomial. El proceso busca la mejor combinación de parámetros alpha y beta y número de inserciones que satisfaga los criterios de cobertura efectiva y frecuencia efectiva mínima (FEM) especificados por el usuario. La función calcula la cobertura acumulada para individuos que han visto el anuncio FEM o más veces.

## Usage

```
optimizar_dc(
  Pob,
  FEM,
  cob_efectiva,
  A1,
  max_inserciones,
  tolerancia = 0.05,
  step_A = 0.1,
  step_B = 0.1,
  batch_size = 1e+06,
```

```

    min_soluciones = 10,
    error_aceptable = 0.01
)

```

### Arguments

Pob	Tamaño de la población
FEM	Frecuencia efectiva mínima (FEM, número mínimo de impactos por persona)
cob_efectiva	Número objetivo de personas a alcanzar con FEM o más contactos
A1	Audiencia tras la primera inserción
max_inserciones	Número de inserciones máximo a considerar
tolerancia	Margen de error permitido en las soluciones (default: 0.05)
step_A	Incremento para búsqueda del parámetro alpha (default: 0.1)
step_B	Incremento para búsqueda del parámetro beta (default: 0.1)
batch_size	Tamaño del lote para procesamiento (default: 1000000)
min_soluciones	Número mínimo de soluciones para parar (default: 10)
error_aceptable	Error aceptable como proporción (default: 0.01)

### Details

La función realiza los siguientes pasos:

1. Validación de parámetros de entrada y dependencias
2. Cálculo de valores objetivo normalizados y tolerancias
3. Generación de combinaciones de parámetros (alpha, beta, n)
4. Cálculo de distribuciones Beta-Binomiales por lotes
5. Suma acumulada de probabilidades para k mayor o igual que FEM
6. Filtrado de resultados según criterios especificados
7. Cálculo de coeficientes R1 y R2 para soluciones válidas
8. Selección de mejor solución y generación de distribución final

### Value

Una lista con los siguientes componentes:

- mejores\_combinaciones: Data frame con todas las combinaciones válidas de parámetros, incluyendo:
  - n: Número de inserciones
  - x: Frecuencia efectiva mínima
  - alpha: Parámetro alpha del modelo
  - beta: Parámetro beta del modelo
  - R1: Proporción de personas alcanzadas tras la primera inserción
  - R2: Proporción de personas alcanzadas tras la segunda inserción
  - prob: Probabilidad acumulada (FEM o más contactos)
  - distancia\_objetivo: Diferencia absoluta con respecto al objetivo

- mejores\_combinaciones\_top\_10: Las 10 mejores combinaciones según criterios
- data: Data frame con la distribución de contactos final, incluyendo:
  - inserciones: Número de inserciones
  - d\_probabilidad: Distribución de probabilidad
  - dc\_acumulada: Distribución acumulada
- alpha: Valor óptimo seleccionado para alpha
- beta: Valor óptimo seleccionado para beta
- n\_optimo: Número óptimo de inserciones

### Note

Esta función difiere de `optimizar_d` en que:

- Utiliza FEM en lugar de FE
- Calcula coberturas acumuladas (personas que ven el anuncio FEM o más veces)
- La optimización considera la suma de probabilidades para k mayor o igual que FEM

### References

Leckenby, J. D., & Boyd, M. M. (1984). An improved beta binomial reach/frequency model for magazines. *Current Issues and Research in Advertising*, 7(1), 1-24.

### See Also

[optimizar\\_d](#) para optimización con frecuencia efectiva exacta [calc\\_R1\\_R2](#) para detalles sobre el cálculo de coeficientes de duplicación

### Examples

```
# Ejemplo 1: Optimización para cobertura acumulada
resultado1 <- optimizar_dc(
  Pob = 1000000,      # Población de 1 millón
  FEM = 3,            # Frecuencia efectiva mínima de 3
  cob_efectiva = 600000, # Objetivo: 600,000 personas con 3+ contactos
  A1 = 450000,        # Primera audiencia: 450,000
  max_inserciones = 8  # Máximo 8 inserciones
)

# Ejemplo 2: Caso con mayor precisión
resultado2 <- optimizar_dc(
  Pob = 500000,
  FEM = 4,
  cob_efectiva = 250000,
  A1 = 200000,
  max_inserciones = 10,
  tolerancia = 0.03,   # Menor tolerancia
  step_A = 0.05,      # Pasos más pequeños
  step_B = 0.05,
  min_soluciones = 15  # Más soluciones requeridas
)

# Comparar probabilidades acumuladas vs objetivo
print(data.frame(
```

```

Objetivo = resultado2$cob_efectiva,
Logrado = resultado2$mejores_combinaciones[1, "prob"],
Error = abs(resultado2$mejores_combinaciones[1, "distancia_objetivo"])
))

```

---

optimize_media_plan	<i>Optimización de planes de medios con restricciones mediante procesamiento por lotes</i>
---------------------	--

---

## Description

Implementa un algoritmo de optimización para encontrar la combinación óptima de soportes publicitarios que maximiza la cobertura para una Frecuencia Efectiva Mínima (FEM) determinada, respetando restricciones presupuestarias. El algoritmo utiliza un enfoque de procesamiento por lotes (batches) y permite elegir entre el modelo de Sainsbury, Binomial o Expansión Canónica para el cálculo de coberturas y distribución de contactos.

## Usage

```

optimize_media_plan(
  soportes_df,
  FEM,
  objetivo_cobertura,
  presupuesto_max,
  tolerancia_presupuesto = 0.1,
  poblacion_total = 4.7e+07,
  tam_batch = 5,
  soportes_vetados = NULL,
  modelo = c("sainsbury", "binomial", "canex"),
  usar_audiencia_util = FALSE,
  correlaciones = NULL,
  truncation_order = 2
)

```

## Arguments

soportes_df	Data frame con columnas: soportes (character), audiencias (numeric), tarifas (numeric). Si se usa audiencia útil, debe incluir también índices_utilidad (numeric)
FEM	Frecuencia Efectiva Mínima requerida (número entero positivo)
objetivo_cobertura	Cobertura objetivo a alcanzar (porcentaje)
presupuesto_max	Presupuesto máximo disponible
tolerancia_presupuesto	Desviación permitida sobre el presupuesto máximo (por defecto 0.10 = 10%)
poblacion_total	Tamaño de la población objetivo (por defecto 47000000)

tam_batch	Tamaño de los lotes para procesamiento (por defecto 5)
soportes_vetados	Vector de caracteres con nombres de soportes a excluir (opcional)
modelo	Modelo a utilizar: "sainsbury", "binomial" o "canex" (por defecto "sainsbury")
usar_audiencia_util	Logical indicando si usar audiencia útil (TRUE) o bruta (FALSE, por defecto)
correlaciones	Matriz de correlaciones entre soportes (requerida solo para modelo "canex")
truncation_order	Orden de truncación para el modelo de expansión canónica (por defecto 2)

## Details

El algoritmo de optimización sigue los siguientes pasos:

1. Preprocesamiento de datos:
  - Filtrado de soportes vetados
  - Cálculo de audiencias útiles si corresponde
  - Cálculo de ratios de eficiencia
  - Validación de matriz de correlaciones para modelo canex
2. Procesamiento por lotes:
  - División en batches para optimizar el proceso
  - Evaluación de combinaciones que cumplen FEM mínima
  - Verificación de restricción presupuestaria
  - Cálculo de cobertura mediante el modelo seleccionado

Los modelos disponibles son:

- Sainsbury: Considera la duplicación entre soportes como producto de probabilidades
- Binomial: Asume una probabilidad media de exposición para todos los soportes
- Canex: Modelo de expansión canónica que considera correlaciones entre soportes

## Value

Una lista conteniendo:

- exito: Logical indicando si se encontró solución factible
- cobertura\_alcanzada: Porcentaje de cobertura logrado
- coste\_total: Coste total del plan
- soportes\_seleccionados: Data frame con los soportes del plan óptimo
- plan\_completo: Vector binario indicando soportes seleccionados
- objetivo\_alcanzado: Logical indicando si se alcanzó el objetivo
- evaluacion: Lista con evaluaciones de presupuesto, cobertura y FEM
- distribucion: Lista con distribución de contactos y acumulada
- soportes\_vetados: Vector de soportes excluidos
- tipo\_audiencia: Tipo de audiencia usada ("bruta" o "util")
- modelo\_usado: Modelo utilizado para los cálculos

**See Also**

[calc\\_sainsbury](#) para el modelo de Sainsbury [calc\\_binomial](#) para el modelo Binomial [canonical\\_expansion\\_model](#) para el modelo de Expansión Canónica

**Examples**

```
# Datos de ejemplo
example_data <- data.frame(
  soportes = c("Medio1", "Medio2", "Medio3"),
  audiencias = c(1000000, 800000, 600000),
  tarifas = c(5000, 4000, 3000)
)

# Ejemplo con modelo Sainsbury
result_sainsbury <- optimize_media_plan(
  soportes_df = example_data,
  FEM = 2,
  objetivo_cobertura = 50,
  presupuesto_max = 100000,
  modelo = "sainsbury"
)

# Ejemplo con modelo Binomial
result_binomial <- optimize_media_plan(
  soportes_df = example_data,
  FEM = 2,
  objetivo_cobertura = 50,
  presupuesto_max = 100000,
  modelo = "binomial"
)

# Ejemplo con modelo Canex
correlaciones <- matrix(
  c(1.000, 0.085, 0.075,
    0.085, 1.000, 0.080,
    0.075, 0.080, 1.000),
  nrow = 3,
  byrow = TRUE
)

result_canex <- optimize_media_plan(
  soportes_df = example_data,
  FEM = 2,
  objetivo_cobertura = 50,
  presupuesto_max = 100000,
  modelo = "canex",
  correlaciones = correlaciones
)
```

## Description

Implementa un algoritmo de optimización para encontrar la combinación óptima de soportes publicitarios que maximiza la cobertura para una Frecuencia Efectiva Mínima (FEM) determinada, respetando restricciones presupuestarias. El algoritmo utiliza un enfoque de procesamiento por lotes (batches) y permite elegir entre el modelo de Sainsbury o el Binomial para el cálculo de coberturas y distribución de contactos. Permite la exclusión de soportes específicos y maneja restricciones presupuestarias con tolerancia configurable. Adicionalmente, permite trabajar con audiencias brutas o audiencias útiles (considerando índices de utilidad).

## Usage

```
optimize_media_sb(
  soportes_df,
  FEM,
  objetivo_cobertura,
  presupuesto_max,
  tolerancia_presupuesto = 0.1,
  poblacion_total = 4.7e+07,
  tam_batch = 5,
  soportes_vetados = NULL,
  modelo = c("sainsbury", "binomial"),
  usar_audiencia_util = FALSE
)
```

## Arguments

soportes_df	Data frame con columnas: soportes (character), audiencias (numeric), tarifas (numeric). Si se usa audiencia útil, debe incluir también índices_utilidad (numeric)
FEM	Frecuencia Efectiva Mínima requerida (número entero positivo)
objetivo_cobertura	Cobertura objetivo a alcanzar (porcentaje)
presupuesto_max	Presupuesto máximo disponible
tolerancia_presupuesto	Desviación permitida sobre el presupuesto máximo (por defecto 0.10 = 10%)
poblacion_total	Tamaño de la población objetivo (por defecto 47000000)
tam_batch	Tamaño de los lotes para procesamiento (por defecto 5)
soportes_vetados	Vector de caracteres con nombres de soportes a excluir (opcional)
modelo	Modelo a utilizar para el cálculo de cobertura: "sainsbury" o "binomial" (por defecto "sainsbury")
usar_audiencia_util	Logical indicando si usar audiencia útil (TRUE) o bruta (FALSE, por defecto)

## Details

El algoritmo de optimización sigue los siguientes pasos:

1. Preprocesamiento de datos:

- Filtrado de soportes vetados
  - Cálculo de audiencias útiles si corresponde ( $\text{audiencia} * \text{índice\_utilidad}$ )
  - Cálculo de ratios de eficiencia ( $\text{audiencia/tarifa}$  o  $\text{audiencia útil/tarifa}$ )
  - Ordenación por eficiencia
2. Procesamiento por lotes:
- División en batches para optimizar el proceso
  - Evaluación de combinaciones que cumplen FEM mínima
  - Verificación de restricción presupuestaria
  - Cálculo de cobertura mediante el modelo seleccionado
3. Optimización iterativa:
- Búsqueda de mejores soluciones en cada batch
  - Actualización progresiva de la mejor solución
  - Criterio de parada por objetivo alcanzado

Los modelos disponibles son:

- Sainsbury: Considera la duplicación entre soportes como el producto de las probabilidades individuales
- Binomial: Asume una probabilidad media de exposición para todos los soportes

## Value

Una lista conteniendo:

- `exit`: Logical indicando si se encontró solución factible
- `cobertura_alcanzada`: Porcentaje de cobertura logrado
- `coste_total`: Coste total del plan
- `soportes_seleccionados`: Data frame con los soportes del plan óptimo. Si se usa audiencia útil, incluye columnas adicionales para audiencia útil e índices de utilidad
- `plan_completo`: Vector binario indicando soportes seleccionados
- `objetivo_alcanzado`: Logical indicando si se alcanzó el objetivo
- `distribucion`: Lista con distribución de contactos y acumulada
- `soportes_vetados`: Vector de soportes excluidos encontrados
- `soportes_no_encontrados`: Vector de soportes vetados no localizados

## References

Aldás Manzano, J. (1998). Modelos de determinación de la cobertura y la distribución de contactos en la planificación de medios publicitarios impresos. Tesis doctoral, Universidad de Valencia, España.

## See Also

[calc\\_sainsbury](#) para el modelo de Sainsbury [calc\\_binomial](#) para el modelo Binomial



## Examples

```
# Ejemplo con audiencia bruta y modelo Sainsbury

datos <- readr::read_csv(file = "data.csv", show_col_types = FALSE)

resultado_bruto <- optimize_media_sb(
  soportes_df = datos,
  FEM = 2,
  objetivo_cobertura = 50,
  presupuesto_max = 100000,
  modelo = "sainsbury",
  usar_audiencia_util = FALSE
)

# Ejemplo con audiencia útil y modelo Binomial
datos_util <- data.frame(
  soportes = c("Medio1", "Medio2", "Medio3"),
  audiencias = c(1000000, 800000, 600000),
  tarifas = c(50000, 40000, 30000),
  indices_utilidad = c(1.2, 1.1, 0.9)
)

resultado_util <- optimize_media_sb(
  soportes_df = datos_util,
  FEM = 2,
  objetivo_cobertura = 50,
  presupuesto_max = 100000,
  modelo = "binomial",
  usar_audiencia_util = TRUE
)
```

---

plot\_grp\_metricas

*Visualización de GRPs y métricas relacionadas por soporte*

---

## Description

Genera un gráfico de puntos para comparar soportes publicitarios según GRPs y coste total, contactos y coste/GRP. El gráfico muestra la relación entre el coste por GRP, los contactos totales y el coste total de cada soporte, utilizando un sistema de burbujas con colores distintivos para cada soporte.

## Usage

```
plot_grp_metricas(
  audiencias,
  inserciones,
  precios,
  nombres,
  pob_total,
  titulo = "Comparación de Soportes Publicitarios"
)
```

**Arguments**

audiencias	Vector numérico con las audiencias de cada soporte
inserciones	Vector numérico del número de inserciones por soporte
precios	Vector numérico con el precio por inserción de cada soporte
nombres	Character vector con los nombres de los soportes
pob_total	Tamaño de la población objetivo
titulo	Character. Título del gráfico (opcional)

**Value**

Un objeto ggplot2 que representa el gráfico de burbujas

**Examples**

```
# Ejemplo básico con tres soportes
plot_grp_metricas(
  audiencias = c(300000, 400000, 200000),
  inserciones = c(3, 2, 4),
  precios = c(1000, 1500, 800),
  nombres = c("Marca", "As", "20 Minutos"),
  pob_total = 1000000,
  titulo = "Análisis de Soportes Deportivos"
)
```

---

run\_aud\_util\_explorer *Explorador de Audiencia Útil*

---

**Description**

Aplicación Shiny para el análisis de audiencias brutas y útiles con diferentes criterios demográficos.

**Usage**

```
run_aud_util_explorer()
```

**Details**

La aplicación permite:

- Configurar una audiencia bruta con distribuciones de sexo, edad y nivel socioeconómico
- Analizar la audiencia útil mediante filtros demográficos
- Visualizar distribuciones mediante gráficos de barras
- Calcular estadísticas relevantes de la audiencia

**Parámetros de Configuración**

- Tamaño de audiencia
- Distribución por sexo (porcentajes)
- Grupos de edad seleccionables
- Niveles socioeconómicos (porcentajes)

---

`run_beta_binomial_explorer`*Función de Masa de Probabilidad Beta-Binomial*

---

**Description**

Calcula la función de masa de probabilidad de la distribución beta-binomial para un conjunto dado de parámetros.

**Usage**`run_beta_binomial_explorer()`**Arguments**

x	Número de éxitos
n	Número de ensayos
alpha	Parámetro de forma alpha de la distribución beta
beta	Parámetro de forma beta de la distribución beta

**Details**

La función implementa la fórmula:

$$P(X = k) = \binom{n}{k} \frac{B(k + \alpha, n - k + \beta)}{B(\alpha, \beta)}$$

---

`run_reach_converg_explorer`*Explorador de Convergencia de la Cobertura*

---

**Description**

Aplicación Shiny para el análisis de la convergencia de la cobertura.

**Usage**`run_reach_converg_explorer()`**Details**

La aplicación permite:

- Configurar un plan de medios aplicando el modelo Beta-Binomial
- Analizar la evolución de la cobertura acumulada e incremental
- Analizar la distribución de contactos (y acumulada)
- Visualizar distribuciones mediante gráficos de líneas
- Calcular estadísticas relevantes de la audiencia

**Parámetros de Configuración**

- Tamaño de población
- Parámetros de forma de la distribución Beta-Binomial
- Máximo número de contactos a mostrar
- Umbral de convergencia

# Index

calc\_beta\_binomial, [4](#), [7](#), [12](#), [14](#), [16](#), [17](#), [19](#),  
[21](#)  
calc\_binomial, [6](#), [6](#), [12](#), [14](#), [16](#), [19](#), [21](#), [30](#), [32](#)  
calc\_cpm, [3](#), [8](#), [10](#)  
calc\_grps, [3](#), [9](#), [9](#)  
calc\_hofmans, [6](#), [7](#), [11](#), [16](#), [19](#), [21](#)  
calc\_MBBD, [13](#)  
calc\_metheringham, [6](#), [7](#), [12](#), [14](#), [15](#), [19](#), [21](#)  
calc\_R1\_R2, [16](#), [24](#), [27](#)  
calc\_sainsbury, [6](#), [7](#), [12](#), [14](#), [16](#), [17](#), [18](#), [21](#),  
[30](#), [32](#)  
calcular\_metricas\_medios, [2](#)  
canonical\_expansion\_model, [19](#), [30](#)  
  
optimizar\_d, [23](#), [27](#)  
optimizar\_dc, [25](#)  
optimize\_media\_plan, [28](#)  
optimize\_media\_sb, [30](#)  
  
plot\_grp\_metricas, [33](#)  
  
run\_aud\_util\_explorer, [34](#)  
run\_beta\_binomial\_explorer, [35](#)  
run\_reach\_converg\_explorer, [35](#)