

Package ‘mediaPlanR’

October 28, 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.

Manual <https://github.com/majesus/mediaPlanR/mediaPlanR_manual.pdf>

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>>

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports stats,
extraDistr,
shiny,
bslib (>= 0.5.0),
ggplot2,
dplyr,
readxl,
readr,
DT,
janitor,
ggrepel,
factoextra,
microbenchmark,
Deriv,
doBy

Suggests knitr,
rmarkdown,
testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

calc_beta_binomial	2
calc_binomial	4
calc_cpm	5
calc_grps	7
calc_metherringham	8
calc_R1_R2	10
calc_sainsbury	11
mediaPlanR	13
optimizar_d	14
optimizar_dc	16
run_beta_binomial_explorer	18

Index	19
--------------	-----------

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 cobertura y distribución de contactos (y acumulada) en planes de medios. Este modelo considera la heterogeneidad en la probabilidad de exposición entre individuos. la Beta-Binomial mezcla estos dos pasos: primero modela la probabilidad de éxito usando la distribución beta de parámetros alpha y beta, y luego utiliza esa probabilidad en una distribución binomial para contar cuántos éxitos se obtienen. Esto es útil cuando la probabilidad de éxito no es conocida de antemano y puede variar entre los individuos. Los parámetros alpha y beta nos permiten ajustar la forma de la distribución para que refleje mejor la incertidumbre que tenemos sobre 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)

See Also

[calc_sainsbury](#) para estimaciones con el modelo de Sainsbury [calc_binomial](#) para estimaciones con la distribución Binomial [calc_R1_R2](#) para el cálculo de coeficientes de duplicación

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) usando modelo Binomial</i>
---------------	--

Description

Implementa un modelo Binomial para calcular la cobertura y distribución de contactos (y acumulada) en un plan de medios. Este modelo asume independencia entre soportes (duplicación aleatoria), y homogeneidad de los soportes e individuos, y utiliza una probabilidad media de exposición (p). La acumulación de las audiencias es un suceso aleatorio. Finalmente, las probabilidades de exposición son estacionarias respecto al tiempo. Las hipótesis aquí expuestas, llevan pues a que la probabilidad de exposición a distintas inserciones en diferentes soportes, sea equivalente a la de distintas inserciones en un soporte hipotético “promedio” cuya audiencia sea la media simple de las audiencias de cada soporte.

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 Binomial 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

See Also

[calc_sainsbury](#) para estimaciones con el modelo de Sainsbury [calc_beta_binomial](#) para estimaciones con la distribución Beta-Binomial

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"
)
```


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 (A2) del hipotético soporte medio. El modelo de Metheringham (1964) se basa en que los individuos tienen probabilidades heterogéneas e independientes de exposición a los soportes, siguiendo una distribución beta; esto convierte la exposición en un proceso de Bernoulli. Además, asume que los soportes son homogéneos, por lo que usa la media de las audiencias para modelar la exposición. La duplicación de audiencias no se considera aleatoria, y al tratar los soportes como homogéneos, la acumulación de m inserciones se modela como una distribución beta-binomial, simplificando el problema de duplicación entre soportes.

Usage

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

Arguments

audiencias	Vector numérico con las audiencias de cada soporte
inserciones	Vector numérico con el número de inserciones por soporte
vector_duplicacion	Vector numérico con valores de duplicación entre soportes
ayuda	Lógico. 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)
- `matriz_oportunidades`: Matriz de oportunidades de contacto
- `vector_oportunidades`: Vector de oportunidades de contacto

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

See Also

[calc_sainsbury](#) para estimaciones con el modelo de Sainsbury [calc_binomial](#) para estimaciones con el modelo Binomial [calc_beta_binomial](#) para estimaciones con el modelo Beta-Binomial

Examples

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

# Mostrar solo la guía de uso
calc_metheringham(
  audiencias = NULL,
  inserciones = NULL,
  vector_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

Cálculo de cobertura y distribución de contactos (y acumulada) según modelo de Sainsbury

Description

Implementa el modelo de Sainsbury simplificado para calcular la cobertura y la distribución de contactos para un conjunto de soportes publicitarios. Este modelo considera la independencia de los soportes (duplicación aleatoria), la homogeneidad de los individuos y la heterogeneidad de los soportes para una estimación más precisa de la cobertura total y la frecuencia de contactos. La probabilidad de que un individuo resulte expuesto al soporte i , vendrá dado por la audiencia del soporte i dividido por la población. Mientras que por la hipótesis de duplicación aleatoria, la probabilidad de exposición continuará siendo una variable bemouilli, pero 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
3. Distribución de contactos acumulada (expuestos al menos i veces)

El proceso incluye:

- Conversión de audiencias a probabilidades
- Cálculo de todas 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

See Also

[calc_binomial](#) para estimaciones con la distribución Binomial [calc_beta_binomial](#) para estimaciones con la distribución Beta-Binomial

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)
```

mediaPlanR

Herramientas para Planificación de Medios

Description

Conjunto de funciones y aplicaciones para planificación de medios.

Details

mediaPlanR: Herramientas para Planificación de Medios

Author(s)

Manuel Sánchez-Franco <majesus@us.es>

References

CRAN

See Also

Useful links:

- <<https://github.com/majesus/mediaPlanR>>

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 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 = 0.025,
  step_B = 0.025
)
```

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)

Details

La función realiza los siguientes pasos:

1. Genera combinaciones de parámetros (alpha, beta) dentro de rangos especificados
2. Calcula distribuciones Beta-Binomiales para cada combinación
3. Filtra resultados según criterios de cobertura y frecuencia
4. Calcula coeficientes R1 y R2
5. Genera visualizaciones de la distribución resultante

Value

Una lista con los siguientes componentes:

- mejores_combinaciones: Data frame con todas las combinaciones válidas de parámetros, incluyendo:
 - x: Número de contactos
 - 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
- mejores_combinaciones_top_10: Las 10 mejores combinaciones según criterios y valores establecidos
- data: Data frame con la distribución de contactos
- alpha: Valor seleccionado para alpha
- beta: Valor seleccionado para beta

Note

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

- alpha: Cuando el valor de alpha aumenta (manteniendo beta constante), se produce una asimetría hacia valores más altos de p (es decir, el éxito es más probable). Esto implica que alpha efectivamente influye en la asimetría, pero en combinación con beta.
- beta: Cuando beta aumenta (y alpha se mantiene constante), se produce una asimetría hacia valores más bajos de p . En este sentido, beta también afecta la asimetría de la distribución beta.

See Also

[calc_R1_R2](#) para los cálculos de R1 y R2

Examples

```
## Not run:
# Ejemplo básico
resultado <- optimizar_d(
  Pob = 1000000,      # Población de 1 millón
  FE = 3,             # Frecuencia efectiva de 3 contactos
  cob_efectiva = 590000, # Objetivo: alcanzar 590,000 personas
  A1 = 500000,        # Audiencia primera inserción: 500,000
  max_inserciones = 5  # Número de inserciones máximo a considerar: 5
)

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

## End(Not run)
```

optimizar_dc	<i>Optimización de distribución de contactos acumulada mediante modelo Beta-Binomial</i>
--------------	--

Description

Optimiza la distribución de contactos acumulada considerando una frecuencia efectiva mínima (FEM) y calcula los coeficientes de duplicación R1 y R2. La función se enfoca en la distribución acumulada de contactos, lo que permite evaluar el alcance para diferentes niveles mínimos de exposición.

Usage

```
optimizar_dc(  
  Pob,  
  FEM,  
  cob_efectiva,  
  A1,  
  max_inserciones,  
  tolerancia = 0.05,  
  step_A = 0.025,  
  step_B = 0.025  
)
```

Arguments

Pob	Tamaño total de la población
FEM	Frecuencia efectiva mínima requerida (FEM, número mínimo de contactos)
cob_efectiva	Número objetivo de personas a alcanzar al menos FEM
A1	Audiencia del soporte tras la primera inserción
max_inserciones	Número de inserciones máximo a considerar (default: 5)
tolerancia	Margen de error permitido para las soluciones (default: 0.05)
step_A	Incremento para la búsqueda del parámetro alpha (default: 0.025)
step_B	Incremento para la búsqueda del parámetro beta (default: 0.025)

Details

El proceso de optimización sigue estos pasos:

1. Validación de parámetros de entrada y normalización
2. Generación de combinaciones de parámetros (alpha, beta)
3. Cálculo de distribuciones Beta-Binomiales
4. Evaluación de probabilidades acumuladas
5. Filtrado de soluciones según criterios específicos:
 - Cumplimiento de la cobertura efectiva
 - Validación de los coeficientes R1 y R2
 - Ajuste a la audiencia objetivo del primer soporte
6. Generación de visualizaciones y resultados

Value

Una lista con los siguientes componentes:

- mejores_combinaciones: Data frame con todas las combinaciones válidas, incluyendo:
 - x: Número de contactos
 - 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
 - probs_acumuladas: Probabilidades acumuladas
- mejores_combinaciones_top_10: Las 10 mejores combinaciones
- data: Data frame con:
 - inserciones: Número de contactos
 - d_probabilidad: Probabilidad individual
 - dc_probabilidad: Probabilidad acumulada
- alpha: Valor seleccionado para alpha
- beta: Valor seleccionado para beta

Note

La función considera la distribución acumulada de contactos, lo que la hace especialmente útil para:

- Planificación de campañas con objetivos de frecuencia efectiva mínima
- Evaluación de cobertura efectiva en diferentes niveles de exposición
- Optimización de planes de medios con requisitos de frecuencia efectiva mínima específicos

See Also

[optimizar_d](#) para optimización de distribución de contactos [calc_R1_R2](#) para los cálculos de R1 y R2

Examples

```
## Not run:
# Ejemplo de optimización para una campaña
resultado <- optimizar_dc(
  Pob = 1000000,      # Población de 1 millón
  FEM = 3,            # Mínimo 3 contactos
  cob_efectiva = 547657, # Objetivo: alcanzar 547,657 personas
  A1 = 500000,        # Audiencia primera inserción: 500,000
  max_inserciones = 5  # Número de inserciones máximo a considerar: 5
)

# Examinar los resultados
print(head(resultado$mejores_combinaciones))
print(resultado$data)

# Ver parámetros óptimos
cat("Alpha óptimo:", resultado$alpha, "\n")
cat("Beta óptimo:", resultado$beta, "\n")

## End(Not run)
```

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

Description

Ejecuta la aplicación Shiny del explorador Beta Binomial

Usage

```
run_beta_binomial_explorer()
```

Value

Una aplicación Shiny

Index

0,1, [11](#)
1,1, [10](#)
1,2, [10](#)
1,3, [10](#)
2,2, [10](#)
2,3, [10](#)
3,3, [10](#)

calc_beta_binomial, [2](#), [5](#), [10–12](#)
calc_binomial, [3](#), [4](#), [10](#), [12](#)
calc_cpm, [5](#), [8](#)
calc_grps, [6](#), [7](#)
calc_metherringham, [8](#)
calc_R1_R2, [3](#), [10](#), [15](#), [17](#)
calc_sainsbury, [3](#), [5](#), [10](#), [11](#), [11](#)

mediaPlanR, [13](#)
mediaPlanR-package (mediaPlanR), [13](#)

optimizar_d, [14](#), [17](#)
optimizar_dc, [16](#)

run_beta_binomial_explorer, [18](#)