

Package ‘agrifit’

December 26, 2025

Title Nonlinear Model Fitting for Agricultural Experiments

Version 0.1.0

Author Jardel Fialho <jardellfialho@gmail.com>

Maintainer Jardel Fialho <jardellfialho@gmail.com>

Description Fits nonlinear models for agricultural dose-response experiments.

Implements Linear Response Plateau (LRP) and other curve-fitting models with automatic breakpoint detection, multiple responses support, publication-ready plots (ggplot2 + plotly), and model selection by R².

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports ggplot2, plotly, dplyr, tidyr

Suggests testthat (>= 3.0.0)

Config/testthat.edition 3

URL <https://github.com/jardel20/agrifit>

BugReports <https://github.com/jardel20/agrifit/issues>

Contents

ajustar_cubico	2
ajustar_exponencial	3
ajustar_linear	4
ajustar_lrp	5
ajustar_lrp2	8
ajustar_mitscherlich	9
ajustar_potencial	10
ajustar_quadratico	11
ajustar_raiz_quadratica	12
analise_superficie_resposta	13
escolha_modelo_adequado	14
otimizacao_insumos	15
otimizacao_superficie_matrizes	16

Index

18

ajustar_cubico *Ajusta o Modelo Cúbico*

Description

Implementa o ajuste do modelo de regressão cúbica: $Y = b_0 + b_1 * X + b_2 * X^2 + b_3 * X^3$.

Usage

```
ajustar_cubico(dose, ..., verbose = TRUE)
```

Arguments

dose	Vetor numérico com níveis do fator (X).
...	Uma ou mais respostas (vetores Y nomeados). Exemplo: MS = <i>materia_seca</i> .
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Lista com:

resultados Data frame: $b_0, b_1, b_2, b_3, R^2, AIC, BIC, RMSE, p$ -valores, IC, etc. para cada resposta.

modelos Lista dos objetos lm para cada resposta.

equacoes Strings formatadas das equações.

Examples

```
## Not run:
dose <- c(0, 50, 100, 150, 200, 250)
resposta <- c(10, 18, 26, 29, 31, 32)

resultado <- ajustar_cubico(
  dose = dose,
  Y = resposta,
  verbose = TRUE
)
print(resultado$resultados)
print(resultado$equacoes)

## End(Not run)
```

ajustar_exponencial Ajusta o Modelo Exponencial

Description

Implementa o ajuste do modelo de regressão exponencial: $Y = a * \exp(b * X)$. O ajuste é feito usando `nls` (Non-linear Least Squares).

Usage

```
ajustar_exponencial(dose, ..., verbose = TRUE)
```

Arguments

<code>dose</code>	Vetor numérico com níveis do fator (X).
<code>...</code>	Uma ou mais respostas (vetores Y nomeados). Exemplo: <code>MS = materia_seca</code> .
<code>verbose</code>	Lógico. Gera prints detalhados (padrão: <code>TRUE</code>).

Value

Lista com:

- resultados** Data frame: a , b , R^2 , AIC, BIC, RMSE, p-valores, IC, etc. para cada resposta.
- modelos** Lista dos objetos `nls` para cada resposta.
- equacoes** Strings formatadas das equações.

Examples

```
## Not run:  
dose <- c(0, 25, 50, 75, 100, 125)  
resposta <- c(10, 12, 15, 20, 28, 40)  
  
resultado <- ajustar_exponencial(  
  dose = dose,  
  Y = resposta,  
  verbose = TRUE  
)  
  
print(resultado$resultados)  
print(resultado$equacoes)  
  
## End(Not run)
```

ajustar_linear*Ajusta o Modelo Linear Simples***Description**

Implementa o ajuste do modelo de regressão linear simples: $Y = b_0 + b_1 * X$.

Usage

```
ajustar_linear(dose, ..., verbose = TRUE)
```

Arguments

dose	Vetor numérico com níveis do fator (X).
...	Uma ou mais respostas (vetores Y nomeados). Exemplo: MS = <code>materia_seca</code> .
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Lista com:

resultados Data frame: b_0 , b_1 , R^2 , AIC, BIC, RMSE, p-valores, IC, etc. para cada resposta.

modelos Lista dos objetos lm para cada resposta.

equacoes Strings formatadas das equações.

Examples

```
## Not run:
dose <- c(0, 50, 100, 150, 200, 250)
resposta <- c(10, 15, 20, 25, 30, 35)

resultado <- ajustar_linear(
  dose = dose,
  Y = resposta,
  verbose = TRUE
)

print(resultado$resultados)
print(resultado$equacoes)

## End(Not run)
```

ajustar_lrp	<i>Ajusta o Modelo Linear Descontínuo (LRP)</i>
-------------	---

Description

Implementa o ajuste iterativo do modelo Linear Response Plateau (LRP), testando cada ponto de dose como potencial breakpoint e selecionando o modelo com maior R². O LRP combina um segmento linear ($\hat{Y} = b_0 + b_1 X$) com plateau ($\hat{Y} = b_2$), onde $X_i = (b_2 - b_0)/b_1$ define a transição.

Implementa o ajuste iterativo do modelo Linear Response Plateau (LRP), testando cada ponto de dose como potencial breakpoint e selecionando o modelo com maior R². O LRP combina um segmento linear ($\hat{Y} = b_0 + b_1 X$) com plateau ($\hat{Y} = b_2$), onde $X_i = (b_2 - b_0)/b_1$ define a transição.

Usage

```
ajustar_lrp(
  dose,
  ...,
  title = "Ajuste do Modelo Linear Descontínuo (LRP)",
  xlab = "Dose",
  ylab = "Resposta",
  adjustment_color = "Set1",
  legend_position_ggplot = "top",
  dashed = FALSE,
  show_intersection = TRUE,
  verbose = TRUE
)

ajustar_lrp(
  dose,
  ...,
  title = "Ajuste do Modelo Linear Descontínuo (LRP)",
  xlab = "Dose",
  ylab = "Resposta",
  adjustment_color = "Set1",
  legend_position_ggplot = "top",
  dashed = FALSE,
  show_intersection = TRUE,
  verbose = TRUE
)
```

Arguments

dose	Vetor numérico com níveis do fator (X). Mínimo 4 pontos distintos.
...	Uma ou mais respostas (vetores Y nomeados). Exemplo: MS = materia_seca.
title	Título do gráfico (padrão: "Ajuste do Modelo Linear Descontínuo (LRP)").
xlab	Rótulo eixo X (padrão: "Dose").
ylab	Rótulo eixo Y (padrão: "Resposta").
adjustment_color	Paleta RColorBrewer para curvas (padrão: "Set1").

```

legend_position_ggplot
    Posição da legenda no ggplot (padrão: "top").
dashed
    Lógico. Se TRUE, adiciona linhas tracejadas nos breakpoints (padrão: FALSE).
show_intersection
    Lógico. Mostra ponto de intersecção (X) no gráfico (padrão: TRUE).
verbose
    Lógico. Gera gráficos e prints detalhados (padrão: TRUE).

```

Details

Atualizações: Inclui cálculo de AIC, BIC, RMSE e testes de significância para os parâmetros do segmento linear (b_0 e b_1).

Algoritmo iterativo:

1. Testa breakpoints de 3 até n-1
2. Regressão linear pré-breakpoint (l_m)
3. Média pós-breakpoint como plateau (b₂)
4. Calcula X_i, R², AIC, BIC, RMSE total
5. Seleciona melhor R²

Saídas gráficas (verbose=TRUE):

- Pontos observados + curvas ajustadas
- Ponto de intersecção opcional (X)
- Linhas tracejadas opcionais (breakpoint)
- Salva "lrp_multiplo.png" (12x8in, 300dpi)

Dependências: ggplot2, plotly, dplyr, tidyverse

Algoritmo iterativo:

1. Testa breakpoints de 3 até n-1
2. Regressão linear pré-breakpoint (l_m)
3. Média pós-breakpoint como plateau (b₂)
4. Calcula X_i e R² total
5. Seleciona melhor R²

Saídas gráficas (verbose=TRUE):

- Pontos observados + curvas ajustadas
- Ponto de intersecção opcional (X)
- Linhas tracejadas opcionais (breakpoint)
- Salva "lrp_multiplo.png" (12x8in, 300dpi)

Dependências: ggplot2, plotly, dplyr, tidyverse

Value

Lista com:

resultados Data frame: b0, b1, b2, Xi, R², AIC, BIC, RMSE, p-valores, etc. para cada resposta.
modelos Lista dos melhores modelos por resposta.
equacoes Strings formatadas das equações LRP.
nomes_respostas Nomes das respostas processadas.
grafico_ggplot Gráfico ggplot2 (se verbose=TRUE).
grafico_plotly Gráfico plotly interativo (se verbose=TRUE).

Lista com:

resultados Data frame: b0, b1, b2, Xi, R², breakpoint para cada resposta.
modelos Lista dos melhores modelos por resposta.
equacoes Strings formatadas das equações LRP.
nomes_respostas Nomes das respostas processadas.
grafico_ggplot Gráfico ggplot2 (se verbose=TRUE).
grafico_plotly Gráfico plotly interativo (se verbose=TRUE).

Examples

```
## Not run:
dose_P <- c(0.0, 32.5, 65.0, 97.5, 130.0, 195.0, 260.0, 325.0)
MS <- c(6.74, 8.73, 10.89, 12.56, 14.11, 15.21, 15.50, 15.60)

res <- ajustar_lrp(
  dose = dose_P, MS = MS,
  title = "Matéria Seca vs Dose de P",
  xlab = "P (mg/dm³)", ylab = "MS (g/vaso)",
  adjustment_color = "Dark2"
)

print(res$resultados)
print(res$equacoes)

## End(Not run)
## Not run:
dose_P <- c(0.0, 32.5, 65.0, 97.5, 130.0, 195.0)
MS <- c(6.74, 8.73, 10.89, 12.56, 14.11, 15.21)

res <- ajustar_lrp(
  dose = dose_P, MS = MS,
  title = "Matéria Seca vs Dose de P",
  xlab = "P (mg/dm³)", ylab = "MS (g/vaso)",
  adjustment_color = "Dark2"
)

print(res$resultados)
print(res$equacoes)

## End(Not run)
```

ajustar_lrp2

Ajusta o Modelo Linear Descontínuo (LRP)

Description

Implementa o ajuste iterativo do modelo Linear Response Plateau (LRP), testando cada ponto de dose como potencial breakpoint e selecionando o modelo com maior R². O LRP combina um segmento linear ($\hat{Y} = b_0 + b_1 X$) com plateau ($\hat{Y} = b_2$), onde $X_i = (b_2 - b_0)/b_1$ define a transição.

Usage

```
ajustar_lrp2(
  dose,
  ...,
  title = "Ajuste do Modelo Linear Descontínuo (LRP)",
  xlab = "Dose",
  ylab = "Resposta",
  adjustment_color = "Set1",
  legend_position_ggplot = "top",
  dashed = FALSE,
  show_intersection = TRUE,
  verbose = TRUE
)
```

Arguments

dose	Vetor numérico com níveis do fator (X). Mínimo 4 pontos distintos.
...	Uma ou mais respostas (vetores Y nomeados). Exemplo: MS = materia_seca.
title	Título do gráfico (padrão: "Ajuste do Modelo Linear Descontínuo (LRP)").
xlab	Rótulo eixo X (padrão: "Dose").
ylab	Rótulo eixo Y (padrão: "Resposta").
adjustment_color	Paleta RColorBrewer para curvas (padrão: "Set1").
legend_position_ggplot	Posição da legenda no ggplot (padrão: "top").
dashed	Lógico. Se TRUE, adiciona linhas tracejadas nos breakpoints (padrão: FALSE).
show_intersection	Lógico. Mostra ponto de intersecção (X) no gráfico (padrão: TRUE).
verbose	Lógico. Gera gráficos e prints detalhados (padrão: TRUE).

Details

Atualizações: Inclui cálculo de AIC, BIC, RMSE e testes de significância para os parâmetros do segmento linear (b_0 e b_1).

Algoritmo iterativo:

1. Testa breakpoints de 3 até n-1
2. Regressão linear pré-breakpoint (lm)

3. Média pós-breakpoint como plateau (b2)
4. Calcula Xi, R², AIC, BIC, RMSE total
5. Seleciona melhor R²

Saídas gráficas (verbose=TRUE):

- Pontos observados + curvas ajustadas
- Ponto de intersecção opcional (X)
- Linhas tracejadas opcionais (breakpoint)
- Salva "lrp_multiplo.png" (12x8in, 300dpi)

Dependências: ggplot2, plotly, dplyr, tidyr

Value

Lista com:

resultados Data frame: b0, b1, b2, Xi, R², AIC, BIC, RMSE, p-valores, etc. para cada resposta.
modelos Lista dos melhores modelos por resposta.
equacoes Strings formatadas das equações LRP.
nomes_respostas Nomes das respostas processadas.
grafico_ggplot Gráfico ggplot2 (se verbose=TRUE).
grafico_plotly Gráfico plotly interativo (se verbose=TRUE).

Examples

```
## Not run:
dose_P <- c(0.0, 32.5, 65.0, 97.5, 130.0, 195.0, 260.0, 325.0)
MS <- c(6.74, 8.73, 10.89, 12.56, 14.11, 15.21, 15.50, 15.60)

res <- ajustar_lrp2(
  dose = dose_P, MS = MS,
  title = "Matéria Seca vs Dose de P",
  xlab = "P (mg/dm³)", ylab = "MS (g/vaso)",
  adjustment_color = "Dark2"
)
print(res$resultados)
print(res$equacoes)

## End(Not run)
```

ajustar_mitscherlich *Ajusta o Modelo de Mitscherlich*

Description

Implementa o ajuste do modelo de Mitscherlich: $Y = A - B * e^{-cX}$. O ajuste é feito usando nls (Non-linear Least Squares).

Usage

```
ajustar_mitscherlich(dose, ..., verbose = TRUE)
```

Arguments

dose	Vetor numérico com níveis do fator (X).
...	Uma ou mais respostas (vetores Y nomeados). Exemplo: MS = <code>materia_seca</code> .
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Lista com:

- resultados** Data frame: A, B, c, R², AIC, BIC, RMSE, p-valores, IC, etc. para cada resposta.
- modelos** Lista dos objetos `nls` para cada resposta.
- equacoes** Strings formatadas das equações.

Examples

```
## Not run:
dose <- c(0, 50, 100, 150, 200, 250)
resposta <- c(5, 12, 18, 22, 25, 27)

resultado <- ajustar_mitscherlich(
  dose = dose,
  Y = resposta,
  verbose = TRUE
)

print(resultado$resultados)
print(resultado$equacoes)

## End(Not run)
```

ajustar_potencial *Ajusta o Modelo Potencial*

Description

Implementa o ajuste do modelo de regressão potencial: $Y = a * X^b$. O ajuste é feito usando `nls` (Non-linear Least Squares).

Usage

```
ajustar_potencial(dose, ..., verbose = TRUE)
```

Arguments

dose	Vetor numérico com níveis do fator (X).
...	Uma ou mais respostas (vetores Y nomeados). Exemplo: MS = <code>materia_seca</code> .
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Lista com:

- resultados** Data frame: a, b, R², AIC, BIC, RMSE, p-valores, IC, etc. para cada resposta.
- modelos** Lista dos objetos `nls` para cada resposta.
- equacoes** Strings formatadas das equações.

Examples

```
## Not run:
dose <- c(1, 10, 20, 50, 100, 200)
resposta <- c(5, 12, 18, 28, 38, 50)

resultado <- ajustar_potencial(
  dose = dose,
  Y = resposta,
  verbose = TRUE
)

print(resultado$resultados)
print(resultado$equacoes)

## End(Not run)
```

ajustar_quadratico *Ajusta o Modelo Quadrático*

Description

Implementa o ajuste do modelo de regressão quadrática: $Y = b_0 + b_1 * X + b_2 * X^2$.

Usage

```
ajustar_quadratico(dose, ..., verbose = TRUE)
```

Arguments

- | | |
|----------------------|---|
| <code>dose</code> | Vetor numérico com níveis do fator (X). |
| <code>...</code> | Uma ou mais respostas (vetores Y nomeados). Exemplo: <code>MS = materia_seca</code> . |
| <code>verbose</code> | Lógico. Gera prints detalhados (padrão: <code>TRUE</code>). |

Value

Lista com:

- resultados** Data frame: $b_0, b_1, b_2, R^2, AIC, BIC, RMSE, p\text{-valores}, IC$, etc. para cada resposta.
- modelos** Lista dos objetos `lm` para cada resposta.
- equacoes** Strings formatadas das equações.

Examples

```
## Not run:
dose <- c(0, 50, 100, 150, 200, 250)
resposta <- c(10, 18, 24, 28, 30, 31)

resultado <- ajustar_quadratico(
  dose = dose,
  Y = resposta,
  verbose = TRUE
)

print(resultado$resultados)
print(resultado$equacoes)

## End(Not run)
```

ajustar_raiz_quadratica

Ajusta o Modelo Raiz Quadrático

Description

Implementa o ajuste do modelo de regressão raiz quadrática: $Y = b_0 + b_1 * \sqrt{X} + b_2 * X$.

Usage

```
ajustar_raiz_quadratica(dose, ..., verbose = TRUE)
```

Arguments

dose	Vetor numérico com níveis do fator (X).
...	Uma ou mais respostas (vetores Y nomeados). Exemplo: MS = materia_seca.
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Lista com:

resultados Data frame: b0, b1, b2, R², AIC, BIC, RMSE, p-valores, IC, etc. para cada resposta.

modelos Lista dos objetos lm para cada resposta.

equacoes Strings formatadas das equações.

Examples

```
## Not run:
dose <- c(0, 25, 50, 100, 150, 200)
resposta <- c(10, 16, 20, 26, 30, 33)

resultado <- ajustar_raiz_quadratica(
  dose = dose,
  Y = resposta,
```

```

    verbose = TRUE
  )

print(resultado$resultados)
print(resultado$equacoes)

## End(Not run)

```

analise_superficie_resposta*Ajusta o Modelo Quadrático de Superfície de Resposta (2 Fatores)***Description**

Implementa o ajuste do modelo quadrático completo para dois fatores (X1 e X2): $Y = b_0 + b_1X_1 + b_2X_2 + b_{11}X_1^2 + b_{22}X_2^2 + b_{12}X_1X_2$.

Usage

```
analise_superficie_resposta(X1, X2, ..., verbose = TRUE)
```

Arguments

X1	Vetor numérico do Fator 1.
X2	Vetor numérico do Fator 2.
...	Uma ou mais respostas (vetores Y nomeadas). Exemplo: Prod = producao.
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Lista com:

resultados Data frame: b0, b1, b2, b11, b22, b12, R², AIC, BIC, RMSE, p-valores, IC, Ponto Estacionário, etc. para cada resposta.

modelos Lista dos objetos lm para cada resposta.

equacoes Strings formatadas das equações.

graficos Lista de gráficos de superfície de resposta 3D para cada resposta.

Examples

```

## Not run:
X1 <- c(-1, -1, 1, 1, 0, 0, 0, 0, 0)
X2 <- c(-1, 1, -1, 1, 0, 0, 0, 0, 0)
prod <- c(50, 60, 55, 70, 65, 65, 65, 66, 64)

resultado <- analise_superficie_resposta(
  X1 = X1,
  X2 = X2,
  Producao = prod,
  verbose = TRUE
)

```

```
)
print(resultado$resultados)
print(resultado$equacoes)
# resultado$graficos$Producao # Para visualizar a superfície 3D

## End(Not run)
```

escolha_modelo_adequado*Escolha do Modelo Adequado***Description**

Consolida e compara os resultados de diferentes modelos de regressão (Linear, Quadrático, etc.) com base em métricas de ajuste como AIC, BIC e R².

Usage

```
escolha_modelo_adequado(..., verbose = TRUE)
```

Arguments

- ... Resultados de ajuste de modelos (listas retornadas pelas funções `ajustar_*`).
- verbose Lógico. Gera prints detalhados (padrão: TRUE).

Value

Data frame consolidado com os resultados de AIC, BIC e R² para todos os modelos e respostas.

Examples

```
## Not run:
dose <- c(0, 50, 100, 150, 200, 250)
resposta <- c(10, 18, 24, 28, 30, 31)

# Ajusta diferentes modelos
modelo_linear <- ajustar_linear(dose, Y = resposta, verbose = FALSE)
modelo_quad <- ajustar_quadratico(dose, Y = resposta, verbose = FALSE)
modelo_cubico <- ajustar_cubico(dose, Y = resposta, verbose = FALSE)

# Compara os modelos
comparacao <- escolha_modelo_adequado(
  modelo_linear,
  modelo_quad,
  modelo_cubico,
  verbose = TRUE
)
print(comparacao)

## End(Not run)
```

otimizacao_insumos *Otimização de Uso de Insumos (DMET e DMEE)*

Description

Calcula a Dose de Máxima Eficiência Técnica (DMET) e a Dose de Máxima Eficiência Econômica (DMEE) para modelos de regressão quadrática ($Y = b_0 + b_1X + b_2X^2$).

Usage

```
otimizacao_insumos(  
  modelo_quadratico,  
  preco_insumo,  
  preco_produto,  
  verbose = TRUE  
)
```

Arguments

modelo_quadratico	Objeto de resultado da função <code>ajustar_quadratico</code> .
preco_insumo	Preço unitário do insumo (dose).
preco_produto	Preço unitário do produto (resposta).
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Data frame com os resultados de otimização (DMET, Y_max, DMEE, Y_DMEE) para cada resposta.

Examples

```
## Not run:  
dose <- c(0, 50, 100, 150, 200, 250)  
resposta <- c(10, 18, 24, 28, 30, 31)  
  
# Ajusta modelo quadrático  
modelo_quad <- ajustar_quadratico(dose, Y = resposta, verbose = FALSE)  
  
# Calcula DMET e DMEE  
otimizacao <- optimizacao_insumos(  
  modelo_quadratico = modelo_quad,  
  preco_insumo = 10,  
  preco_produto = 50,  
  verbose = TRUE  
)  
  
print(otimizacao)  
  
## End(Not run)
```

otimizacao_superficie_matrizes

*Otimização de Superfície de Respostas por Álgebra de Matrizes
(DMEE)*

Description

Calcula a Dose de Máxima Eficiência Econômica (DMEE) para o modelo quadrático de Superfície de Resposta (2 Fatores) usando Álgebra de Matrizes.

Usage

```
otimizacao_superficie_matrizes(
  modelo_superficie,
  preco_insumo_X1,
  preco_insumo_X2,
  preco_produto,
  verbose = TRUE
)
```

Arguments

modelo_superficie	Objeto de resultado da função analise_superficie_resposta.
preco_insumo_X1	Preço unitário do insumo X1.
preco_insumo_X2	Preço unitário do insumo X2.
preco_produto	Preço unitário do produto (resposta).
verbose	Lógico. Gera prints detalhados (padrão: TRUE).

Value

Lista com:

resultados Data frame com os resultados de otimização (X1_DMEE, X2_DMEE, Y_DMEE) para cada resposta.

graficos Lista de gráficos 3D de superfície de resposta com pontos marcados de DMEE para cada resposta.

Examples

```
## Not run:
X1 <- c(-1, -1, 1, 1, 0, 0, 0, 0, 0)
X2 <- c(-1, 1, -1, 1, 0, 0, 0, 0, 0)
prod <- c(50, 60, 55, 70, 65, 65, 65, 66, 64)

# Ajusta modelo de superfície
superficie <- analise_superficie_resposta(
  X1 = X1, X2 = X2,
  Producao = prod,
```

```
    verbose = FALSE
)

# Calcula DMEE
otimizacao <- otimizacao_superficie_matrizes(
  modelo_superficie = superficie,
  preco_insumo_X1 = 20,
  preco_insumo_X2 = 25,
  preco_produto = 100,
  verbose = TRUE
)

print(otimizacao)

## End(Not run)
```

Index

ajustar_cubico, 2
ajustar_exponencial, 3
ajustar_linear, 4
ajustar_lrp, 5
ajustar_lrp2, 8
ajustar_mitscherlich, 9
ajustar_potencial, 10
ajustar_quadratico, 11
ajustar_raiz_quadratico, 12
analise_superficie_resposta, 13

escolha_modelo_adequado, 14

otimizacao_insumos, 15
otimizacao_superficie_matrizes, 16