

Package ‘GREENREG’

September 21, 2026

Title Tool for Statistical and Environmental Analysis

Version 0.1.1

Description Provides a set of accessible and automated functions to apply statistical models such as Simple Linear Regression (RLS, from the Spanish 'Regresión Lineal Simple'), Multiple Linear Regression (RLM, from the Spanish 'Regresión Lineal Múltiple'), Generalized Linear Models (GLM), and time series analysis through Autoregressive Integrated Moving Average (ARIMA) models. Designed to support teaching at the Universidad Autónoma Chapingo, it facilitates results interpretation and assumption validation through automatic graphical diagnostics. Developed as part of an undergraduate thesis at the Universidad Autónoma Chapingo, under the supervision of Dr. Julio César Buendía Espinoza (thesis advisor), with the participation of the thesis committee: Diego Ernesto Lira González (secretary), Israel Lerma Serna (member), Juan Uriel Avelar Roblero (alternate), and Elisa del Carmen Martínez Ochoa (alternate). Methods for regression and time series are based on Montgomery et al. (2021, ISBN:978-1119570141) and Box & Jenkins (1970, ISBN:978-0816211043).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5)

Imports ggplot2, ggrepel, stats, lmttest

Suggests pROC, broom, car

NeedsCompilation no

Author Dayron Jared Bazan Guzman [aut, cre],
Julio César Buendía Espinoza [aut, ths] (affiliation: Universidad
Autónoma Chapingo, secondary_email: jcbuendiae@hotmail.com),
Diego Ernesto Lira González [ctb] (Secretario del comité de tesis,
Universidad Autónoma Chapingo),
Israel Lerma Serna [ctb] (Vocal del comité de tesis, Universidad
Autónoma Chapingo),
Juan Uriel Avelar Roblero [ctb] (Suplente del comité de tesis,

Universidad Autónoma Chapingo),
Elisa del Carmen Martínez Ochoa [ctb] (Suplente del comité de tesis,
Universidad Autónoma Chapingo)

Maintainer Dayron Jared Bazan Guzman <dbazanguzman@gmail.com>

Repository CRAN

Date/Publication 2026-09-21 10:00:46 UTC

Contents

analisis_datos	2
datos_anomalia_temperatura	4
datos_nivel_presa	5
datos_rendimiento_maiz	6
datos_roya_cafe	7
datos_tala_ilegal_poisson	8
logistico	9
modelo_ar	11
modelo_arima	14
modelo_arma	18
modelo_ma	21
reg_poisson	24
rlm	26
rls	28
Index	30

analisis_datos	<i>Asistente Interactivo de Analisis de Datos</i>
----------------	---

Description

Esta es la funcion principal y punto de entrada del paquete. Actua como un consultor automatizado que guia al usuario desde la carga de datos hasta la generacion del codigo final, seleccionando la metodologia adecuada segun la naturaleza de la variable objetivo.

Usage

```
analisis_datos(data, variable_objetivo, verbose = TRUE)
```

Arguments

- data Data frame o base de datos que contiene las variables.
- variable_objetivo Cadena de texto con el nombre exacto de la columna que deseas analizar (ej. "Ventas").
- verbose Logico. Si es TRUE, despliega los menus. Por defecto es TRUE.

Value

Devuelve de forma invisible un objeto de clase ggplot2. La funcion se utiliza principalmente por su efecto secundario de imprimir codigo sugerido en la consola y generar visualizaciones.

Como funciona este Asistente paso a paso?

1. **Escaneo y Validacion:** Analiza la base de datos para asegurar que la variable objetivo existe. Ademas, realiza un test interno de autocorrelacion para verificar si tiene sentido aplicar modelos de Series de Tiempo (evitando analisis espurios).
2. **Menu Interactivo:** Despliega opciones en la consola basandose en los datos detectados, preguntando al usuario que enfoque desea (Regresion Clasica, Modelos Generalizados o Series de Tiempo).
3. **Generacion de Estrategias (El "Motor"):**
 - *Para Regresion y GLM:* Crea dos versiones del codigo.
 - (a) **Modelo Full:** Incluye todas las variables disponibles.
 - (b) **Modelo Optimizado:** Ejecuta internamente un algoritmo "Stepwise" basado en AIC para limpiar el modelo y quedarse solo con los predictores relevantes.
 - *Para Series de Tiempo:* Ejecuta una "Busqueda en Rejilla" (Grid Search) probando combinaciones de parametros (p, d, q). Identifica automaticamente si la serie necesita diferenciacion (ARIMA), si es puramente autoregresiva (AR), de media movil (MA) o mixta (ARMA), y genera el codigo para la funcion especifica.

Guia de la Grafica Exploratoria Generada

Dependiendo de la opcion elegida, la funcion genera una visualizacion preliminar diferente para entender los datos antes de modelar:

Escenario 1: Regresion Lineal (Histograma + Gauss) Muestra un histograma azul de tus datos superpuesto con una curva roja (Campana de Gauss teorica). *Objetivo:* Evaluar visualmente si la variable respuesta se parece a una distribucion Normal (supuesto clave de la regresion lineal).

Escenario 2: GLM (Barras o Histograma) Detecta el tipo de variable:

- *Si es Binaria (0/1):* Grafico de barras naranja. Ayuda a ver si las clases estan desbalanceadas (ej. muchos "0" y pocos "1").
- *Si es Conteo:* Histograma verde. Muestra la frecuencia de eventos (tipico de Poisson).

Escenario 3: Serie de Tiempo (Linea de Tendencia) Grafica la evolucion de la variable a lo largo del indice temporal. *Objetivo:* Identificar tendencias (subidas/bajadas a largo plazo) o ciclos repetitivos antes de ajustar el modelo.

Examples

```
data(datos_rendimiento_maiz)
analisis_datos(datos_rendimiento_maiz, "rendimiento_maiz_ton_ha")
```

`datos_anomalia_temperatura`*Anomalías de temperatura media diaria*

Description

Conjunto de datos que muestra las anomalías (diferencias respecto a la media histórica) de la temperatura media diaria registradas a lo largo del tiempo. Este dataset puede utilizarse para ilustrar análisis de series de tiempo y modelos autorregresivos o ARIMA.

Usage

`datos_anomalia_temperatura`

Format

Un data frame con 730 observaciones y 2 variables:

fecha Fecha del registro (AAAA-MM-DD)

anomia_c Anomalia de temperatura respecto al promedio historico (C)

Details

Los datos fueron contruidos de manera simulada a partir de patrones reales observados en bases climaticas del INEGI y de la CONAGUA. La creacion propia de esta base permite garantizar que los valores sean adecuados para fines didacticos y compatibles con las funciones del paquete **GREEN-REG**.

Source

Elaboracion propia con fines didacticos a partir de informacion publica de INEGI y CONAGUA.

Examples

```
data("datos_anomalia_temperatura")
modelo_AR <- modelo_ar(datos_anomalia_temperatura$anomia_c, p = 2)
modelo_AR
plot(modelo_AR)
```

datos_nivel_presa	<i>Niveles de agua en presas de Mexico</i>
-------------------	--

Description

Conjunto de datos que presenta los niveles de almacenamiento de agua registrados en distintas presas y fechas, util para analisis de series de tiempo y modelos ARIMA.

Usage

```
datos_nivel_presa
```

Format

Un data frame con 1825 observaciones y 2 variables:

fecha Fecha del registro (AAAA-MM-DD)

nivel_m Nivel de almacenamiento de la presa (metros)

Details

Los datos fueron simulados a partir de patrones observados en reportes de CONAGUA, manteniendo coherencia temporal y realismo. Se generaron con fines didacticos para demostrar la aplicacion de modelos AR, MA y ARIMA en **GREENREG**.

Source

Elaboracion propia con fines didacticos basada en patrones de datos de CONAGUA.

Examples

```
data("datos_nivel_presa")
ts_data <- datos_nivel_presa$nivel_m
modelo_ARIMA <- modelo_arima(ts_data, p = 1, d = 1, q = 1)
print(modelo_ARIMA)
plot(modelo_ARIMA)
```

`datos_rendimiento_maiz`*Rendimiento de maiz bajo distintos niveles de fertilizacion*

Description

Conjunto de datos que presenta el rendimiento del maiz en funcion de diferentes variables climaticas y de manejo del cultivo. Este dataset es util para ejemplificar modelos de regresion lineal simple y multiple.

Usage

`datos_rendimiento_maiz`

Format

Un data frame con 250 observaciones y 4 variables:

rendimiento_maiz_ton_ha Rendimiento del cultivo de maiz (ton/ha)

precipitacion_mm Precipitacion registrada durante el ciclo agricola (mm)

temperatura_c Temperatura media (C)

fertilizante_kg_ha Cantidad de fertilizante aplicado (kg/ha)

Details

Los datos fueron generados de forma controlada, tomando como referencia distribuciones y rangos reales reportados por SIAP e INEGI. Esta base se diseno especificamente para probar las funciones de regresion del paquete **GREENREG**.

Source

Elaboracion propia con fines didacticos tomando referencia de datos del SIAP e INEGI.

Examples

```
data(datos_rendimiento_maiz)
modelo_rlm <- rlm(rendimiento_maiz_ton_ha ~., data = datos_rendimiento_maiz)
modelo_rlm
plot(modelo_rlm)
```

datos_roya_cafe	<i>Presencia de roya del cafe en diferentes localidades</i>
-----------------	---

Description

Base de datos simulada que presenta la presencia/ausencia de roya del cafe junto con variables ambientales y de manejo agronomico relevantes. Es util para ejercicios de regresion logistica.

Usage

```
datos_roya_cafe
```

Format

Un data frame con 300 observaciones y 4 variables:

presencia_roya Presencia (1) o ausencia (0) de roya del cafe

humedad_relativa_pct Humedad relativa promedio (%)

altitud_msnm Altitud de la finca (m s.n.m.)

manejo_sombra Porcentaje de cobertura de sombra en la plantacion

Details

Los datos fueron generados de forma controlada utilizando parametros similares a los observados por SIAP y estudios agronomicos del INIFAP. Esta base fue creada para ser utilizada en funciones de regresion logistica del paquete **GREENREG**.

Source

Elaboracion propia con fines didacticos basada en informacion de SIAP e INIFAP.

Examples

```
data(datos_roya_cafe)
modelo_log <- logistico(presencia_roya ~ humedad_relativa_pct +
                        altitud_msnm + manejo_sombra,
                        data = datos_roya_cafe)
```

Registros de tala ilegal en distintas regiones

Conjunto de datos que contiene el numero de eventos de tala ilegal observados junto con variables explicativas relacionadas con la densidad poblacional, superficie forestal y esfuerzo de vigilancia. Ideal para ejemplificar la aplicacion de modelos de regresion Poisson.

```
datos_tala_ilegal_poisson
```

Un data frame con 200 observaciones y 4 variables:

dias_vigilancia Numero de dias de vigilancia reportados

Los datos fueron generados de manera simulada con base en valores reportados por CONAFOR y SEMARNAT. Esto permite realizar ejercicios reproducibles y seguros sin exponer datos sensibles o confidenciales.

Elaboracion propia con fines didacticos tomando referencia de CONAFOR e INEGI.

[illegible]

logistico

*Regresion Logistica Binaria (MLG Logit)***Description**

Esta funcion ajusta un Modelo Lineal Generalizado (GLM) de la familia binomial para predecir la probabilidad de un evento dicotomico (Si/No, 0/1). A diferencia de la regresion lineal, el motor de calculo estima los coeficientes mediante Maxima Verosimilitud (MLE) y ofrece un diagnostico especializado en clasificacion.

Genera una secuencia de 9 graficas para validar el modelo de regresion logistica.

Usage

```
logistico(formula, data, umbral = 0.5)

## S3 method for class 'logistico_greenreg'
print(x, ...)

## S3 method for class 'logistico_greenreg'
plot(x, ...)
```

Arguments

formula	Una formula de R (ej. presencia_plaga ~ humedad + altitud).
data	Un data.frame con las observaciones.
umbral	Punto de corte para clasificar como "Evento (1)" (por defecto 0.5).
x	Objeto de clase logistico_greenreg.
...	Argumentos adicionales.

Details

El flujo interno de la funcion se divide en cinco etapas fundamentales:

1. **Ajuste Logit:** Transforma la variable respuesta mediante la funcion de enlace $\text{logit}(p) = \ln(p/(1 - p))$, permitiendo que la probabilidad se mantenga entre 0 y 1.
2. **Conversion de Impacto:** Los coeficientes estimados se transforman automaticamente a **Odds Ratios (OR)** mediante e^{β} . Esto permite concluir cuanto multiplica cada variable la posibilidad de que ocurra el evento.
3. **Evaluacion de Bondad (McFadden):** Calcula el Pseudo- R^2 comparando la verosimilitud del modelo ajustado contra un modelo nulo (sin predictores).
4. **Metricas de Clasificacion:** Genera la Matriz de Confusion basada en el umbral especificado y calcula la Precision Global (Accuracy).
5. **Diagnostico de Influencia:** Evalua si existen observaciones que deforman las probabilidades mediante el calculo de la Distancia de Cook y el Apalancamiento (Leverage).

Value

Un objeto de clase "logistico_greenreg", que consiste en una lista con los siguientes componentes:

- **formula:** La formula de R aplicada al modelo.
- **modelo:** El objeto de clase glm original con el ajuste binomial.
- **variables:** Una lista que contiene el nombre de la variable respuesta (y).
- **coeficientes:** Matriz con las estimaciones de los coeficientes, errores estandar, estadisticos z y p-valores.
- **odds_ratios:** Tabla con los Odds Ratios (OR) calculados, sus intervalos de confianza y p-valores asociados.
- **metricas:** Una lista que incluye el Pseudo-R2 de McFadden, el Criterio de Informacion de Akaike (AIC) y la precision global (*Accuracy*).
- **confusion:** Matriz de confusion que cruza los valores reales frente a los predichos segun el umbral especificado.
- **predicciones:** Un data.frame con los valores observados, probabilidades predichas, clasificacion final y el apalancamiento (*leverage*).
- **diag_leverage:** Lista con informacion diagnostica sobre el apalancamiento y deteccion de puntos extremos.
- **notas:** Lista de mensajes interpretativos sobre la bondad de ajuste y la precision.
- **supuestos:** Un vector de caracteres con las validaciones tecnicas (semaforos) sobre significancia, multicolinealidad e influencia.

Guia de las 9 Graficas de Diagnostico (plot)

Al ejecutar `plot(modelo)`, se despliega una secuencia de validacion predictiva:

- 1. Capacidad de Separacion** Muestra si el modelo realmente distingue entre grupos. Buscamos "montanas" de densidad que no se traslapen.
- 2. Matriz de Confusion Visual** Mapa de calor que resume aciertos (diagonal) y errores (falsos positivos/negativos).
- 3. Odds Ratios (Forest Plot)** Visualiza la importancia de las variables. Los puntos a la derecha del 1 son factores de riesgo; a la izquierda, protectores.
- 4. Curva de Calibracion** Compara la probabilidad que "dice" el modelo contra la frecuencia real observada. Debe seguir la diagonal.
- 5. Curva ROC (AUC)** Mide la potencia global de diagnostico. Un AUC de 1.0 es una clasificacion perfecta; 0.5 es puro azar.
- 6. Distancia de Cook** Detecta casos atipicos en la clasificacion que tienen un peso excesivo en los coeficientes.
- 7. Trade-off de Umbrales** Permite ver como cambia la Sensibilidad y Especificidad al mover el punto de corte.
- 8. Residuos de Pearson** Busca patrones sistematicos en el error. Idealmente, la linea roja debe mantenerse cerca del cero.
- 9. Leverage vs Residuos** Identifica observaciones con combinaciones de variables X muy raras que "jalonean" las probabilidades.

Examples

```
data(datos_roya_cafe)
modelo_log <- logistico(presencia_roya ~ humedad_relativa_pct +
                        altitud_msnm + manejo_sombra,
                        data = datos_roya_cafe)
```

modelo_ar

Modelo Autorregresivo (AR)

Description

Esta funcion ajusta un modelo de series de tiempo donde el valor actual (Y_t) se explica mediante una combinacion lineal de sus propios valores pasados ($Y_{t-1}, Y_{t-2}, \dots$). Es la herramienta estandar para analizar la "persistencia" o "memoria" de un fenomeno a traves del tiempo.

Genera una secuencia de 9 graficas de diagnostico para validar el modelo AR.

Esta funcion ajusta un modelo de series de tiempo donde el valor actual (Y_t) se explica mediante una combinacion lineal de sus propios valores pasados ($Y_{t-1}, Y_{t-2}, \dots$). Es la herramienta estandar para analizar la "persistencia" o "memoria" de un fenomeno a traves del tiempo.

Genera una secuencia de 9 graficas de diagnostico para validar el modelo AR.

Usage

```
modelo_ar(x, p = 1, nivel_confianza = 0.95)

## S3 method for class 'ar_greenreg'
print(x, ...)

## S3 method for class 'ar_greenreg'
plot(x, ...)

modelo_ar(x, p = 1, nivel_confianza = 0.95)

## S3 method for class 'ar_greenreg'
print(x, ...)

## S3 method for class 'ar_greenreg'
plot(x, ...)
```

Arguments

x	Para modelo_ar(): vector numerico o serie de tiempo (ts) de entrada, sin valores NA. Para print()/plot(): objeto de clase "ar_greenreg" devuelto por modelo_ar().
p	Orden del modelo (numero de rezagos a considerar). Por defecto es 1.

nivel_confianza Nivel de significancia para las bandas de error (default 0.95).
 ... Argumentos adicionales.

Details

El flujo interno de la funcion se divide en cinco etapas diagnosticas:

1. **Estimacion (MLE):** Utiliza Maxima Verosimilitud para encontrar los coeficientes ϕ (Phi) que mejor describen la dependencia temporal.
2. **Analisis de Estabilidad:** Calcula las raices inversas del polinomio caracteristico. Un modelo AR solo es valido si es "Estacionario", es decir, si las raices se encuentran dentro del circulo unitario.
3. **Inferencia de Parametros:** Calcula errores estandar y p-valores para determinar cuantos rezagos (lags) son realmente significativos.
4. **Auditoria de Residuos:** Evalua si el error restante es "Ruido Blanco" mediante la prueba de Ljung-Box (independencia) y Shapiro-Wilk (normalidad).
5. **Traduccion Didactica:** Genera un reporte que explica si el modelo "explota" (inestable) o si ha capturado exitosamente la memoria de la serie.

El flujo interno de la funcion se divide en cinco etapas diagnosticas:

1. **Estimacion (MLE):** Utiliza Maxima Verosimilitud para encontrar los coeficientes ϕ (Phi) que mejor describen la dependencia temporal.
2. **Analisis de Estabilidad:** Calcula las raices inversas del polinomio caracteristico. Un modelo AR solo es valido si es "Estacionario", es decir, si las raices se encuentran dentro del circulo unitario.
3. **Inferencia de Parametros:** Calcula errores estandar y p-valores para determinar cuantos rezagos (lags) son realmente significativos.
4. **Auditoria de Residuos:** Evalua si el error restante es "Ruido Blanco" mediante la prueba de Ljung-Box (independencia) y Shapiro-Wilk (normalidad).
5. **Traduccion Didactica:** Genera un reporte que explica si el modelo "explota" (inestable) o si ha capturado exitosamente la memoria de la serie.

Value

Un objeto de clase "ar_greenreg" que contiene:

- modelo: Objeto arima original.
- orden_p: El orden del modelo ajustado.
- coeficientes: Matriz con estimaciones, errores estandar y p-valores.
- estabilidad: Lista con las raices inversas y diagnostico de estacionariedad.
- diagnosticos: Resultados de las pruebas de Ljung-Box, Shapiro-Wilk y AIC.
- notas: Reporte interpretado sobre estabilidad y ruido blanco.
- data_plot: Dataframe listo para visualización avanzada.

Un objeto de clase "ar_greenreg" que contiene:

- modelo: Objeto arima original.
- orden_p: El orden del modelo ajustado.
- coeficientes: Matriz con estimaciones, errores estandar y p-valores.
- estabilidad: Lista con las raices inversas y diagnostico de estacionariedad.
- diagnosticos: Resultados de las pruebas de Ljung-Box, Shapiro-Wilk y AIC.
- notas: Reporte interpretado sobre estabilidad y ruido blanco.
- data_plot: Dataframe listo para visualización avanzada.

Guia de las 9 Graficas de Diagnostico (plot)

Al ejecutar `plot(modelo)`, se despliega una auditoria temporal completa:

- 1. Inspeccion Visual** Muestra la serie bruta con su media historica. Ayuda a detectar tendencias que el modelo AR no puede manejar por si solo.
- 2. Ajuste Temporal** Superpone el modelo sobre la realidad. La sombra roja indica la incertidumbre (Intervalo de Confianza).
- 3. Residuos en el Tiempo** Un "electrocardiograma" del error. No debe presentar ondas, ciclos ni tendencias.
- 4. Circulo Unitario** Grafica matematica critica. Si los rombos rojos salen del circulo, el modelo es inestable y no sirve para pronosticar.
- 5. ACF de Residuos** Busca correlaciones remanentes. Las barras deben ser pequenas para confirmar que el error es ruido aleatorio.
- 6. PACF de Residuos** Ayuda a confirmar si el orden p seleccionado fue suficiente para limpiar la dependencia directa.
- 7. Normalidad (Q-Q Plot)** Valida que los choques aleatorios sigan una distribucion normal, requisito para que los intervalos de confianza sean exactos.
- 8. Precision (Real vs Ajustado)** Muestra la correlacion directa entre la prediccion y el dato observado.
- 9. Estacionariedad Movil** Analiza si la media y la varianza cambian a lo largo de la serie. Una linea verde horizontal confirma un modelo robusto.

Al ejecutar `plot(modelo)`, se despliega una auditoria temporal completa:

- 1. Inspeccion Visual** Muestra la serie bruta con su media historica. Ayuda a detectar tendencias que el modelo AR no puede manejar por si solo.
- 2. Ajuste Temporal** Superpone el modelo sobre la realidad. La sombra roja indica la incertidumbre (Intervalo de Confianza).
- 3. Residuos en el Tiempo** Un "electrocardiograma" del error. No debe presentar ondas, ciclos ni tendencias.
- 4. Circulo Unitario** Grafica matematica critica. Si los rombos rojos salen del circulo, el modelo es inestable y no sirve para pronosticar.
- 5. ACF de Residuos** Busca correlaciones remanentes. Las barras deben ser pequenas para confirmar que el error es ruido aleatorio.

- 6. PACF de Residuos** Ayuda a confirmar si el orden p seleccionado fue suficiente para limpiar la dependencia directa.
- 7. Normalidad (Q-Q Plot)** Valida que los choques aleatorios sigan una distribucion normal, requisito para que los intervalos de confianza sean exactos.
- 8. Precision (Real vs Ajustado)** Muestra la correlacion directa entre la prediccion y el dato observado.
- 9. Estacionariedad Movil** Analiza si la media y la varianza cambian a lo largo de la serie. Una linea verde horizontal confirma un modelo robusto.

Examples

```
data("datos_anomalia_temperatura")
modelo_AR <- modelo_ar(datos_anomalia_temperatura$anomalia_c, p = 2)
modelo_AR
plot(modelo_AR)
```

```
data("datos_anomalia_temperatura")
modelo_AR <- modelo_ar(datos_anomalia_temperatura$anomalia_c, p = 2)
modelo_AR
plot(modelo_AR)
```

modelo_arima

Modelo Integrado de Media Movil Autorregresivo (ARIMA)

Description

Esta funcion ajusta el modelo mas versatil para series de tiempo univariadas: el ARIMA(p , d , q). Combina la memoria de largo plazo (AR), la estabilizacion mediante diferenciacion (I) para eliminar tendencias, y la respuesta a choques aleatorios de corto plazo (MA).

Genera un panel visual para evaluar el ajuste del modelo.

Esta funcion ajusta el modelo mas versatil para series de tiempo univariadas: el ARIMA(p , d , q). Combina la memoria de largo plazo (AR), la estabilizacion mediante diferenciacion (I) para eliminar tendencias, y la respuesta a choques aleatorios de corto plazo (MA).

Genera un panel visual para evaluar el ajuste del modelo.

Usage

```
modelo_arima(x, p = 1, d = 1, q = 1, include_mean = TRUE)

## S3 method for class 'arima_greenreg'
print(x, ...)

## S3 method for class 'arima_greenreg'
plot(x, ...)
```

```

modelo_arima(x, p = 1, d = 1, q = 1, include_mean = TRUE)

## S3 method for class 'arima_greenreg'
print(x, ...)

## S3 method for class 'arima_greenreg'
plot(x, ...)

```

Arguments

x	Para modelo_arima(): vector numerico o serie de tiempo (ts) de entrada, sin valores NA. Para print()/plot(): objeto de clase "arima_greenreg" devuelto por modelo_arima().
p	Orden Autorregresivo (memoria del pasado).
d	Orden de Integracion (numero de diferenciaciones).
q	Orden de Media Movil (memoria de los errores).
include_mean	Logico. Incluir constante? (Suele ser FALSE si $d > 0$).
...	Argumentos adicionales.

Details

El flujo interno de la funcion ejecuta una auditoria de cinco fases:

1. **Diferenciacion (Fase I):** Si $d > 0$, el modelo resta los valores consecutivos para eliminar la tendencia y lograr que la serie sea "Estacionaria" (promedio constante).
2. **Estimacion ARMA:** Sobre la serie ya estabilizada, estima los parametros ϕ (AR) y θ (MA) mediante Maxima Verosimilitud.
3. **Doble Validacion de Estructura:** Analiza las raices del polinomio caracteristico para asegurar que la parte AR sea **Estable** y la parte MA sea **Invertible**.
4. **Auditoria de Residuos:** Verifica que el error final sea "Ruido Blanco" mediante las pruebas de Ljung-Box (independencia) y Shapiro-Wilk (normalidad).
5. **Diagnostico Dinamico:** Evalua si la media y la varianza se mantienen estables a lo largo del tiempo despues de la integracion.

El flujo interno de la funcion ejecuta una auditoria de cinco fases:

1. **Diferenciacion (Fase I):** Si $d > 0$, el modelo resta los valores consecutivos para eliminar la tendencia y lograr que la serie sea "Estacionaria" (promedio constante).
2. **Estimacion ARMA:** Sobre la serie ya estabilizada, estima los parametros ϕ (AR) y θ (MA) mediante Maxima Verosimilitud.
3. **Doble Validacion de Estructura:** Analiza las raices del polinomio caracteristico para asegurar que la parte AR sea **Estable** y la parte MA sea **Invertible**.
4. **Auditoria de Residuos:** Verifica que el error final sea "Ruido Blanco" mediante las pruebas de Ljung-Box (independencia) y Shapiro-Wilk (normalidad).
5. **Diagnostico Dinamico:** Evalua si la media y la varianza se mantienen estables a lo largo del tiempo despues de la integracion.

Value

Un objeto de clase "arima_greenreg" que contiene:

- **modelo:** El objeto arima original de stats.
- **ordenes:** Vector con los valores de p , d y q .
- **coeficientes_tabla:** Matriz con estimaciones, errores estandar y p-valores.
- **estatus:** Lista con el estado de estabilidad e invertibilidad.
- **diagnosticos:** Resultados de Ljung-Box, Shapiro-Wilk y AIC.
- **data_plot:** Dataframe listo para visualización con ggplot2.
- **data_diff:** Dataframe de la serie diferenciada (si $d > 0$).

Un objeto de clase "arima_greenreg" que contiene:

- **modelo:** El objeto arima original de stats.
- **ordenes:** Vector con los valores de p , d y q .
- **coeficientes_tabla:** Matriz con estimaciones, errores estandar y p-valores.
- **estatus:** Lista con el estado de estabilidad e invertibilidad.
- **diagnosticos:** Resultados de Ljung-Box, Shapiro-Wilk y AIC.
- **data_plot:** Dataframe listo para visualización con ggplot2.
- **data_diff:** Dataframe de la serie diferenciada (si $d > 0$).

Guia de las 10 Graficas de Diagnostico

Al ejecutar `plot(modelo)`, se despliega el diagnostico mas completo del paquete:

- **1. Serie Original:** Visualiza la serie bruta. Fundamental para identificar la tendencia inicial que justifica el valor de d .
- **2. Ajuste Integrado:** Superpone el modelo (púrpura) sobre la realidad, incluyendo bandas de confianza del 95%.
- **3. Serie Diferenciada:** Muestra la serie después de aplicar d . Si el modelo es correcto, esta serie debe verse plana y sin tendencia.
- **4. Circulo Unitario:** Grafica de convergencia matemática. Los puntos rojos (AR) y azules (MA) deben estar dentro del círculo gris.
- **5. Residuos en el Tiempo:** Busca patrones de error. Los puntos deben formar una nube aleatoria sin ciclos visibles.
- **6. ACF de Residuos:** Prueba de independencia. Las barras deben estar dentro de los límites para confirmar el Ruido Blanco.
- **7. PACF de Residuos:** Ayuda a identificar si el orden autorregresivo p es suficiente para explicar la inercia.
- **8. Normalidad (Q-Q Plot):** Valida que los errores sigan una distribución gaussiana para asegurar pronósticos confiables.
- **9. Precision (Real vs Ajustado):** Evaluación de la correlación de Pearson entre los datos y la estimación.

- **10. Estacionariedad Final:** Muestra la media y varianza movil. Una franja purpura estable certifica un modelo robusto.

Al ejecutar `plot(modelo)`, se despliega el diagnostico mas completo del paquete:

- **1. Serie Original:** Visualiza la serie bruta. Fundamental para identificar la tendencia inicial que justifica el valor de d .
- **2. Ajuste Integrado:** Superpone el modelo (purpura) sobre la realidad, incluyendo bandas de confianza del 95\
- **3. Serie Diferenciada:** Muestra la serie despues de aplicar d . Si el modelo es correcto, esta serie debe verse plana y sin tendencia.
- **4. Circulo Unitario:** Grafica de convergencia matematica. Los puntos rojos (AR) y azules (MA) deben estar dentro del circulo gris.
- **5. Residuos en el Tiempo:** Busca patrones de error. Los puntos deben formar una nube aleatoria sin ciclos visibles.
- **6. ACF de Residuos:** Prueba de independencia. Las barras deben estar dentro de los limites para confirmar el Ruido Blanco.
- **7. PACF de Residuos:** Ayuda a identificar si el orden autorregresivo p es suficiente para explicar la inercia.
- **8. Normalidad (Q-Q Plot):** Valida que los errores sigan una distribucion gaussiana para asegurar pronosticos confiables.
- **9. Precision (Real vs Ajustado):** Evaluacion de la correlacion de Pearson entre los datos y la estimacion.
- **10. Estacionariedad Final:** Muestra la media y varianza movil. Una franja purpura estable certifica un modelo robusto.

See Also

[arima](#)

[arima](#)

Examples

```
data("datos_nivel_presa")
ts_data <- datos_nivel_presa$nivel_m
modelo_ARIMA <- modelo_arima(ts_data, p = 1, d = 1, q = 1)
print(modelo_ARIMA)
plot(modelo_ARIMA)
```

```
data("datos_nivel_presa")
ts_data <- datos_nivel_presa$nivel_m
modelo_ARIMA <- modelo_arima(ts_data, p = 1, d = 1, q = 1)
print(modelo_ARIMA)
plot(modelo_ARIMA)
```

 modelo_arma

Modelo Autorregresivo de Media Movil (ARMA)

Description

Esta funcion ajusta un modelo hibrido ARMA(p, q) que combina la dependencia de los valores pasados (AutoRegresivo) con la influencia de los errores pasados (Media Movil). Es la herramienta estandar para modelar series de tiempo estacionarias con estructuras de dependencia complejas.

Genera una secuencia de 9 graficas de diagnostico para validar el modelo ARMA.

Esta funcion ajusta un modelo hibrido ARMA(p, q) que combina la dependencia de los valores pasados (AutoRegresivo) con la influencia de los errores pasados (Media Movil). Es la herramienta estandar para modelar series de tiempo estacionarias con estructuras de dependencia complejas.

Genera una secuencia de 9 graficas de diagnostico para validar el modelo ARMA.

Usage

```
modelo_arma(x, p = 1, q = 1, include_mean = TRUE)
```

```
## S3 method for class 'arma_greenreg'
print(x, ...)
```

```
## S3 method for class 'arma_greenreg'
plot(x, ...)
```

```
modelo_arma(x, p = 1, q = 1, include_mean = TRUE)
```

```
## S3 method for class 'arma_greenreg'
print(x, ...)
```

```
## S3 method for class 'arma_greenreg'
plot(x, ...)
```

Arguments

x	Para modelo_arma(): vector numerico o serie de tiempo (ts) de entrada, sin valores NA. Para print()/plot(): objeto de clase "arma_greenreg" devuelto por modelo_arma().
p	Orden Autorregresivo (pasados de la variable).
q	Orden de Media Movil (pasados del error).
include_mean	Logico. Incluir constante/intercepto? Por defecto TRUE.
...	Argumentos adicionales.

Details

El flujo interno de la funcion realiza una auditoria estructural en cinco pasos:

1. **Estimacion Conjunta:** Utiliza Maxima Verosimilitud (MLE) para encontrar simultaneamente los coeficientes ϕ (AR) y θ (MA).
2. **Estabilidad (Parte AR):** Verifica que el componente autorregresivo no sea explosivo. Las raices inversas del polinomio AR deben estar dentro del circulo.
3. **Invertibilidad (Parte MA):** Verifica que el componente de media movil sea univoco. Las raices inversas del polinomio MA deben estar dentro del circulo.
4. **Diagnostico de Residuos:** Evalua si el error resultante es Ruido Blanco mediante el test de Ljung-Box, asegurando que no quede senal sin capturar.
5. **Criterio de Informacion:** Proporciona el AIC para facilitar la comparacion entre diferentes combinaciones de (p, q), buscando el equilibrio entre complejidad y precision.

El flujo interno de la funcion realiza una auditoria estructural en cinco pasos:

1. **Estimacion Conjunta:** Utiliza Maxima Verosimilitud (MLE) para encontrar simultaneamente los coeficientes ϕ (AR) y θ (MA).
2. **Estabilidad (Parte AR):** Verifica que el componente autorregresivo no sea explosivo. Las raices inversas del polinomio AR deben estar dentro del circulo.
3. **Invertibilidad (Parte MA):** Verifica que el componente de media movil sea univoco. Las raices inversas del polinomio MA deben estar dentro del circulo.
4. **Diagnostico de Residuos:** Evalua si el error resultante es Ruido Blanco mediante el test de Ljung-Box, asegurando que no quede senal sin capturar.
5. **Criterio de Informacion:** Proporciona el AIC para facilitar la comparacion entre diferentes combinaciones de (p, q), buscando el equilibrio entre complejidad y precision.

Value

Un objeto de clase "arma_greenreg" que contiene:

- modelo: Objeto arima original (orden p, 0, q).
- ordenes: Vector con los valores de p y q.
- coeficientes_tabla: Matriz con estimaciones, errores estandar y p-valores.
- estatus: Lista con el estado de estabilidad e invertibilidad.
- diagnosticos: Resultados de Ljung-Box, Shapiro-Wilk y AIC.
- data_plot: Dataframe listo para visualizacion con ggplot2.

Un objeto de clase "arma_greenreg" que contiene:

- modelo: Objeto arima original (orden p, 0, q).
- ordenes: Vector con los valores de p y q.
- coeficientes_tabla: Matriz con estimaciones, errores estandar y p-valores.
- estatus: Lista con el estado de estabilidad e invertibilidad.
- diagnosticos: Resultados de Ljung-Box, Shapiro-Wilk y AIC.
- data_plot: Dataframe listo para visualizacion con ggplot2.

Guia de las 9 Graficas de Diagnostico (plot)

Al ejecutar `plot(modelo)`, se despliega una secuencia de validacion mixta:

- 1. Inspeccion de Serie** Muestra la serie bruta con su media. Fundamental para confirmar visualmente la estacionariedad.
- 2. Ajuste Mixto** Superpone el modelo (verde) sobre la realidad. Permite evaluar como interactuan la inercia y los shocks.
- 3. Residuos en el Tiempo** Evaluacion de la aleatoriedad del error. Los puntos deben formar una nube sin patrones.
- 4. Circulo Unitario Dual** Grafica critica exclusiva de ARMA. Muestra simultaneamente la Estabilidad (Puntos Rojos) e Invertibilidad (Rombos Azules).
- 5. ACF de Residuos** Busca autocorrelaciones remanentes. Las barras deben ser minimas para confirmar que el modelo es optimo.
- 6. PACF de Residuos** Ayuda a identificar si el orden autorregresivo (p) es suficiente.
- 7. Normalidad (Q-Q Plot)** Valida que los errores sigan una campana de Gauss, requisito para la validez de los intervalos de confianza.
- 8. Precision (Real vs Ajustado)** Muestra la correlacion directa. Un ARMA exitoso suele tener una alineacion estrecha sobre la diagonal.
- 9. Estacionariedad Movil** Prueba final de estabilidad. Una franja verde horizontal confirma que el modelo no se desplaza en el tiempo.

Al ejecutar `plot(modelo)`, se despliega una secuencia de validacion mixta:

- 1. Inspeccion de Serie** Muestra la serie bruta con su media. Fundamental para confirmar visualmente la estacionariedad.
- 2. Ajuste Mixto** Superpone el modelo (verde) sobre la realidad. Permite evaluar como interactuan la inercia y los shocks.
- 3. Residuos en el Tiempo** Evaluacion de la aleatoriedad del error. Los puntos deben formar una nube sin patrones.
- 4. Circulo Unitario Dual** Grafica critica exclusiva de ARMA. Muestra simultaneamente la Estabilidad (Puntos Rojos) e Invertibilidad (Rombos Azules).
- 5. ACF de Residuos** Busca autocorrelaciones remanentes. Las barras deben ser minimas para confirmar que el modelo es optimo.
- 6. PACF de Residuos** Ayuda a identificar si el orden autorregresivo (p) es suficiente.
- 7. Normalidad (Q-Q Plot)** Valida que los errores sigan una campana de Gauss, requisito para la validez de los intervalos de confianza.
- 8. Precision (Real vs Ajustado)** Muestra la correlacion directa. Un ARMA exitoso suele tener una alineacion estrecha sobre la diagonal.
- 9. Estacionariedad Movil** Prueba final de estabilidad. Una franja verde horizontal confirma que el modelo no se desplaza en el tiempo.

Examples

```
data("datos_anomalia_temperatura")
ts_data <- datos_anomalia_temperatura$anomalia_c
modelo <- modelo_arma(ts_data, p = 1, q = 1)
modelo
plot(modelo)

data("datos_anomalia_temperatura")
ts_data <- datos_anomalia_temperatura$anomalia_c
modelo <- modelo_arma(ts_data, p = 1, q = 1)
modelo
plot(modelo)
```

modelo_ma	<i>Modelo de Media Movil (MA)</i>
-----------	-----------------------------------

Description

Esta funcion ajusta un modelo de series de tiempo donde el valor actual (Y_t) se explica mediante una combinacion lineal de los errores (shocks) pasados (ϵ_{t-1} , ϵ_{t-2} , ...). Es ideal para modelar fenomenos con efectos de memoria corta o impactos transitorios.

Genera una secuencia de 9 graficas de diagnostico para validar el modelo MA.

Esta funcion ajusta un modelo de series de tiempo donde el valor actual (Y_t) se explica mediante una combinacion lineal de los errores (shocks) pasados (ϵ_{t-1} , ϵ_{t-2} , ...). Es ideal para modelar fenomenos con efectos de memoria corta o impactos transitorios.

Genera una secuencia de 9 graficas de diagnostico para validar el modelo MA.

Usage

```
modelo_ma(x, q = 1, include_mean = TRUE)

## S3 method for class 'ma_greenreg'
print(x, ...)

## S3 method for class 'ma_greenreg'
plot(x, ...)

modelo_ma(x, q = 1, include_mean = TRUE)

## S3 method for class 'ma_greenreg'
print(x, ...)

## S3 method for class 'ma_greenreg'
plot(x, ...)
```

Arguments

x	Para modelo_ma(): vector numerico o serie de tiempo (ts) de entrada, sin valores NA. Para print()/plot(): objeto de clase "ma_greenreg" devuelto por modelo_ma().
q	Orden del modelo (numero de choques pasados a considerar). Por defecto es 1.
include_mean	Logico. Incluir el promedio constante de la serie? Por defecto TRUE.
...	Argumentos adicionales.

Details

El flujo interno de la funcion se divide en cinco etapas fundamentales:

1. **Estimacion (MLE):** Utiliza Maxima Verosimilitud para estimar los parametros θ (Theta). A diferencia del AR, el MA requiere algoritmos iterativos ya que los errores no son directamente observables.
2. **Análisis de Invertibilidad:** Calcula las raices inversas del polinomio caracteristico. Un modelo MA debe ser "Invertible" para que el error pueda expresarse como una funcion convergente de los datos observados.
3. **Identificacion Estructural:** Evalua la ACF (Autocorrelacion). En un proceso MA(q), la ACF debe "cortarse" abruptamente despues del lag q.
4. **Auditoria de Residuos:** Verifica que el error remanente sea Ruido Blanco mediante la prueba de Ljung-Box y la distribucion Gaussiana (Shapiro-Wilk).
5. **Traduccion Didactica:** Genera un reporte que explica la persistencia de los choques y valida si el modelo es unico y matematicamente solido.

El flujo interno de la funcion se divide en cinco etapas fundamentales:

1. **Estimacion (MLE):** Utiliza Maxima Verosimilitud para estimar los parametros θ (Theta). A diferencia del AR, el MA requiere algoritmos iterativos ya que los errores no son directamente observables.
2. **Análisis de Invertibilidad:** Calcula las raices inversas del polinomio caracteristico. Un modelo MA debe ser "Invertible" para que el error pueda expresarse como una funcion convergente de los datos observados.
3. **Identificacion Estructural:** Evalua la ACF (Autocorrelacion). En un proceso MA(q), la ACF debe "cortarse" abruptamente despues del lag q.
4. **Auditoria de Residuos:** Verifica que el error remanente sea Ruido Blanco mediante la prueba de Ljung-Box y la distribucion Gaussiana (Shapiro-Wilk).
5. **Traduccion Didactica:** Genera un reporte que explica la persistencia de los choques y valida si el modelo es unico y matematicamente solido.

Value

Un objeto de clase "ma_greenreg", que consiste en una lista con:

- modelo: El objeto arima original (orden 0,0,q).
- orden_q: El orden del modelo ajustado.

- `coeficientes_tabla`: Matriz con estimaciones, errores estandar y p-valores.
- `invertibilidad`: Lista con las raices inversas y diagnostico de invertibilidad.
- `diagnosticos`: Resultados de las pruebas de Ljung-Box, Shapiro-Wilk y AIC.
- `data_plot`: Dataframe listo para visualizacion con ggplot2.

Un objeto de clase "ma_greenreg", que consiste en una lista con:

- `modelo`: El objeto arima original (orden 0,0,q).
- `orden_q`: El orden del modelo ajustado.
- `coeficientes_tabla`: Matriz con estimaciones, errores estandar y p-valores.
- `invertibilidad`: Lista con las raices inversas y diagnostico de invertibilidad.
- `diagnosticos`: Resultados de las pruebas de Ljung-Box, Shapiro-Wilk y AIC.
- `data_plot`: Dataframe listo para visualizacion con ggplot2.

Guia de las 9 Graficas de Diagnostico (plot)

Al ejecutar `plot(modelo)`, se despliega una auditoria de "shocks" temporales:

- 1. Inspeccion Visual** Muestra la serie bruta con su media historica. El modelo MA supone que la serie siempre es atraida hacia este nivel central.
- 2. Ajuste Temporal** Superpone el modelo (azul) sobre la realidad. Permite ver como el modelo "suaviza" los picos de la serie.
- 3. Precision (Real vs Ajustado)** Grafica de dispersion para evaluar la correlacion. Los puntos deben seguir la diagonal roja.
- 4. Circulo Unitario** Prueba de Invertibilidad. Los rombos azules deben estar dentro del circulo para asegurar que el modelo sea unico.
- 5. Residuos en el Tiempo** Evaluacion de aleatoriedad. No debe haber patrones; los errores deben ser impredecibles.
- 6. ACF de Residuos** Prueba critica de Ruido Blanco. Si hay barras largas (numeros rojos), el orden q es insuficiente.
- 7. PACF de Residuos** Complemento de la ACF para verificar que no existan dependencias directas remanentes.
- 8. Normalidad (Q-Q Plot)** Valida que los choques sigan una campana de Gauss, asegurando la validez de los intervalos de confianza.
- 9. Estacionariedad Movil** Confirma que la media y varianza no se desplazan. Vital para asegurar que el modelo MA sea aplicable.

Al ejecutar `plot(modelo)`, se despliega una auditoria de "shocks" temporales:

- 1. Inspeccion Visual** Muestra la serie bruta con su media historica. El modelo MA supone que la serie siempre es atraida hacia este nivel central.
- 2. Ajuste Temporal** Superpone el modelo (azul) sobre la realidad. Permite ver como el modelo "suaviza" los picos de la serie.
- 3. Precision (Real vs Ajustado)** Grafica de dispersion para evaluar la correlacion. Los puntos deben seguir la diagonal roja.

- 4. Circulo Unitario** Prueba de Invertibilidad. Los rombos azules deben estar dentro del circulo para asegurar que el modelo sea unico.
- 5. Residuos en el Tiempo** Evaluacion de aleatoriedad. No debe haber patrones; los errores deben ser impredecibles.
- 6. ACF de Residuos** Prueba critica de Ruido Blanco. Si hay barras largas (numeros rojos), el orden q es insuficiente.
- 7. PACF de Residuos** Complemento de la ACF para verificar que no existan dependencias directas remanentes.
- 8. Normalidad (Q-Q Plot)** Valida que los choques sigan una campana de Gauss, asegurando la validez de los intervalos de confianza.
- 9. Estacionariedad Movil** Confirma que la media y varianza no se desplazan. Vital para asegurar que el modelo MA sea aplicable.

Examples

```
data("datos_anomalia_temperatura")
modelo_MA <- modelo_ma(datos_anomalia_temperatura$anomalia_c, q = 1, include_mean = TRUE)
modelo_MA
plot(modelo_MA)

data("datos_anomalia_temperatura")
modelo_MA <- modelo_ma(datos_anomalia_temperatura$anomalia_c, q = 1, include_mean = TRUE)
modelo_MA
plot(modelo_MA)
```

reg_poisson

Regresion de Poisson (Modelos de Conteo)

Description

Esta funcion ajusta un Modelo Lineal Generalizado (GLM) de la familia Poisson con funcion de enlace logaritmica $\ln(\lambda)$. Esta disenada para modelar variables de respuesta que representan conteos enteros no negativos (0, 1, 2, 3...) y tasas de incidencia en estudios agronomicos, forestales y sociales.

Genera una secuencia de 8 graficas de diagnostico para validar el modelo Poisson.

Usage

```
reg_poisson(formula, data)

## S3 method for class 'poisson_greenreg'
print(x, ...)

## S3 method for class 'poisson_greenreg'
plot(x, ...)
```

Arguments

formula	Una formula de R (ej. numero_frutos ~ riego + fertilizante).
data	Un data.frame con las observaciones (el conteo debe ser entero).
x	Objeto de clase.
...	Argumentos adicionales.

Details

El flujo interno de la funcion se divide en seis etapas criticas:

1. **Ajuste Log-Lineal:** Estima la relacion entre los predictores y el logaritmo de la tasa de eventos (λ). Utiliza Maxima Verosimilitud (MLE) para el ajuste.
2. **Interpretacion de Tasas (IRR):** Los coeficientes originales (β) se exponencian automaticamente e^β para obtener los **Ratios de Tasas de Incidencia**. Esto permite concluir en que porcentaje aumenta o disminuye la frecuencia de eventos por cada unidad de cambio en X.
3. **Diagnostico de Equidispersion:** Calcula el Ratio de Dispersion basado en el estadistico de Pearson (X^2/gl). En una distribucion Poisson perfecta, este valor debe ser 1. Valores > 1.5 indican **Sobredispersion**, sugiriendo que los datos varian mas de lo que el modelo permite.
4. **Inferencia Robusta:** Genera intervalos de confianza para los IRR, permitiendo identificar factores de riesgo ($IRR > 1$) y factores protectores ($IRR < 1$).
5. **Evaluacion de Residuos:** Calcula residuos de devianza y Pearson para identificar patrones de error sistematicos o falta de ajuste en conteos especificos (como el cero).
6. **Traduccion Didactica:** Genera un reporte que advierte si el modelo es valido o si se debe migrar a un modelo Binomial Negativa debido a la dispersion.

Value

Un objeto de clase "poisson_greenreg" que contiene:

- modelo: Objeto glm original de la familia Poisson.
- irr_table: Tabla con los Ratios de Tasas de Incidencia, intervalos y p-valores.
- dispersion: El ratio de dispersion calculado (Pearson / GL).
- metricas: Lista con AIC, Log-Verosimilitud y Devianzas.
- predicciones: Dataframe con valores observados, lambda estimado y diagnosticos (leverage/residuos).

Guia de las 8 Graficas de Diagnostico (plot)

Al ejecutar plot(modelo), se despliega una secuencia de validacion para conteos:

1. **Distribucion de Conteos (Real vs Modelo)** Compara las frecuencias observadas contra las que el modelo "esperaba" predecir. Crucial para detectar el exceso de ceros.
2. **Precision (Predicho vs Real)** Scatterplot entre el Lambda estimado y el conteo real. Los puntos deben seguir la tendencia de la linea roja diagonal.
3. **Importancia de Variables (IRR)** Visualiza el impacto relativo de cada predictor. Los puntos verdes son significativos; si el intervalo cruza el 1, la variable es ruido.

- 4. Verificación de Sobredispersión** Grafica la relación Media-Varianza. Si la línea verde (datos) se eleva sobre la negra (teoría), el supuesto Poisson se rompe.
- 5. Distancia de Cook** Identifica casos atípicos en el conteo que tienen un peso excesivo en la estimación de la tasa promedio.
- 6. Residuos de Devianza** Analiza la calidad del ajuste en escala logarítmica. Busca una nube uniforme sin formas de "U" o de embudo.
- 7. Apalancamiento (Leverage)** Detecta observaciones con valores de X inusuales que controlan la pendiente del modelo.
- 8. Densidad de Ajuste Global** Superpone las curvas de probabilidad del modelo y la realidad para evaluar el solapamiento total.

Examples

```
data(datos_tala_ilegal_poisson)
modelo_poisson <- reg_poisson(numero_talas ~ densidad_poblacion +
                             superficie_forestal_ha + dias_vigilancia,
                             data = datos_tala_ilegal_poisson)

plot(modelo_poisson)
```

rlm

Regresión Lineal Múltiple (RLM)

Description

Esta función ajusta un modelo de regresión lineal con múltiples variables predictoras $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$ mediante Mínimos Cuadrados Ordinarios (OLS). Incluye un robusto motor de diagnóstico de Multicolinealidad (VIF) y pruebas de supuestos con recomendaciones de acción para el investigador.

Genera una secuencia de 10 gráficas para validar el modelo.

Usage

```
rlm(formula, data, nivel_confianza = 0.95)

## S3 method for class 'rlm_greenreg'
print(x, ...)

## S3 method for class 'rlm_greenreg'
plot(x, ...)
```

Arguments

formula	Fórmula modelo.
data	Dataframe con los datos.
nivel_confianza	Nivel de confianza para intervalos (default 0.95).
x	Objeto de clase rlm_greenreg.
...	Argumentos adicionales.

Details

El flujo interno de la funcion se divide en seis etapas:

1. **Validacion de Datos:** Asegura que la formula sea valida y que el dataframe no contenga errores estructurales.
2. **Ajuste (Engine):** Utiliza `stats::lm`. En RLM, el foco principal es el **R2 Ajustado**, que penaliza la inclusion de variables innecesarias que no aportan informacion real.
3. **Deteccion de Multicolinealidad (VIF):** Calcula manualmente el Factor de Inflacion de la Varianza $VIF = 1/(1 - R_j^2)$ para cada predictor. Esto identifica si las variables X estan correlacionadas entre si, lo cual infla los errores estandar y vuelve inestables los coeficientes.
4. **Auditoria de Supuestos:** Ejecuta de forma nativa pruebas de Normalidad (Shapiro), Homocedasticidad (Breusch-Pagan), Independencia (Durbin-Watson) y Linealidad (Rainbow Test).
5. **Traduccion Automatica:** Evalua los estadisticos y genera "Notas". y sugerencias de mejora (ej. transformar variables o eliminar predictores redundantes).
6. **Preparacion de Diagnostico:** Construye un dataframe optimizado para ggplot2 que incluye residuos estandarizados, distancias de Cook y apalancamiento (leverage).

Value

Un objeto de clase "rlm_greenreg" que contiene:

- `modelo`: Objeto `lm` original.
- `variables`: Lista con los nombres de Y y los predictores X.
- `coeficientes`: Matriz con estimaciones, p-valores e intervalos de confianza.
- `vif`: Vector con los valores del Factor de Inflacion de la Varianza (VIF).
- `supuestos`: Resultados de los tests de Shapiro-Wilk (Normalidad), Breusch-Pagan (Homocedasticidad), Durbin-Watson (Independencia) y Rainbow (Linealidad).
- `notas`: Reporte interpretado con recomendaciones de accion.
- `data_plot`: Dataframe optimizado para visualizaciones.

Guia de las 10 Graficas de Diagnostico (plot)

Al ejecutar `plot(modelo)`, se despliega una secuencia educativa de 10 visualizaciones:

1. **Distribucion de Variables (Y, Xi)** Muestra histogramas y densidades de todas las variables. Permite detectar sesgos o datos atipicos antes del ajuste.
2. **Relacion de Variables (Y vs Xi)** Scatterplots individuales de la respuesta contra cada predictor con una curva LOESS para verificar la linealidad visualmente.
3. **Ajuste (Observados vs Ajustados)** Muestra la precision global. Los puntos deben seguir la diagonal de 45 grados.
4. **Q-Q Plot (Normalidad)** Valida si los residuos siguen una distribucion normal. Crucial para la validez de los P-valores.
5. **Homocedasticidad** Residuos vs. Predichos. Busca una "nube de estrellas" sin forma de embudo.

- 6. Independencia** Residuos vs. Orden de recoleccion. Detecta patrones temporales o espaciales en los errores.
- 7. Prediccion vs Realidad (Identidad)** Version simplificada del ajuste para evaluar la correlacion de Pearson.
- 8. Multicolinealidad (VIF)** Grafica de barras exclusiva de RLM. Las barras rojas ($VIF > 10$) indican variables que deben ser eliminadas por redundancia.
- 9. Distancia de Cook** Identifica observaciones influyentes. Barras que superan el umbral $4/n$ indican datos que deforman los resultados.
- 10. Leverage vs Residuos** Cruza la "rareza" de X (Leverage) con el error en Y (Residuo). Detecta puntos que actuan como palancas sobre la recta.

Examples

```
data(datos_rendimiento_maiz)
modelo_rlm <- rlm(rendimiento_maiz_ton_ha ~ ., data = datos_rendimiento_maiz)
print(modelo_rlm)
plot(modelo_rlm)
```

rls

Regresion Lineal Simple (RLS)

Description

Esta funcion ajusta un modelo de regresion lineal $y = \beta_0 + \beta_1 x$ mediante Minimos Cuadrados Ordinarios (OLS, por sus siglas en ingles 'Ordinary Least Squares'). A diferencia de `lm()` estandar, esta funcion devuelve un objeto enriquecido con interpretaciones en lenguaje natural, verificaciones automaticas de supuestos y sugerencias para el investigador.

Genera una secuencia de 9 graficas para validar el modelo.

Usage

```
rls(formula, data, nivel_confianza = 0.95)
```

```
## S3 method for class 'rls_greenreg'
print(x, ...)
```

```
## S3 method for class 'rls_greenreg'
plot(x, ...)
```

Arguments

<code>formula</code>	Una formula de R estandar (ej. <code>rendimiento ~ fertilizante</code>).
<code>data</code>	El data.frame que contiene las variables.
<code>nivel_confianza</code>	Nivel de confianza para los intervalos (por defecto 0.95).
<code>x</code>	Objeto de clase <code>rls_greenreg</code> .
<code>...</code>	Argumentos adicionales.

Details

El flujo interno de la funcion se divide en cinco etapas criticas:

1. **Preparacion:** Valida que los datos sean coherentes y extrae las variables de la formula.
2. **Estimacion:** Calcula los coeficientes, el error estandar residual (σ) y la significancia global (F-test).
3. **Inferencia:** Genera intervalos de confianza para los parametros (β) al nivel especificado.
4. **Auditoria de Supuestos:** Ejecuta pruebas de Shapiro-Wilk (Normalidad), Breusch-Pagan (Homocedasticidad) y Durbin-Watson (Independencia).
5. **Traduccion:** Convierte los estadisticos abstractos en "Notas" de texto que recomiendan acciones especificas (ej. aplicar logaritmos o revisar datos atipicos).

Value

Un objeto de clase "rls_greenreg" que es una lista con:

- modelo: El objeto lm original.
- variables: Nombres de X e Y.
- coeficientes: Matriz con Estimaciones, Errores e Intervalos.
- supuestos: Resultados de los tests de Shapiro, BP y DW.
- notas: Diccionario de interpretaciones y sugerencias de mejora.
- data_plot: Dataframe con residuos e indices de influencia.

Guia de las 9 Graficas de Diagnostico (plot)

Al ejecutar plot(modelo), se despliega una secuencia educativa:

1. **Histogramas Exploratorios** Compara la distribucion de X e Y. Util para detectar sesgos iniciales.
2. **Relacion Original (Scatterplot)** Muestra la nube de puntos pura para confirmar linealidad.
3. **Recta de Regresion** Visualiza el ajuste final con bandas de confianza.
4. **Q-Q Plot (Normalidad)** Grafica los cuantiles de los residuos contra una normal teorica.
5. **Homocedasticidad** Residuos vs. Predichos para verificar varianza constante.
6. **Independencia** Residuos vs. Orden de datos para detectar autocorrelacion.
7. **Precision (Predicho vs Real)** Muestra la correlacion directa entre valores reales y estimados.
8. **Distancia de Cook** Identifica observaciones influyentes o atipicos peligrosos.
9. **Leverage vs Residuos** Diferencia entre errores en Y (Outliers) y valores extremos en X (Apalancamiento).

Examples

```
data(datos_rendimiento_maiz)
modelo <- rls(rendimiento_maiz_ton_ha ~ precipitacion_mm, data = datos_rendimiento_maiz)
print(modelo)
plot(modelo)
```

Index

* **datasets**
 datos_anomalia_temperatura, 4
 datos_nivel_presa, 5
 datos_rendimiento_maiz, 6
 datos_roya_cafe, 7
 datos_tala_ilegal_poisson, 8

analisis_datos, 2
arima, 17

datos_anomalia_temperatura, 4
datos_nivel_presa, 5
datos_rendimiento_maiz, 6
datos_roya_cafe, 7
datos_tala_ilegal_poisson, 8

logistico, 9

modelo_ar, 11
modelo_arima, 14
modelo_arma, 18
modelo_ma, 21

plot.ar_greenreg (modelo_ar), 11
plot.arima_greenreg (modelo_arima), 14
plot.arma_greenreg (modelo_arma), 18
plot.logistico_greenreg (logistico), 9
plot.ma_greenreg (modelo_ma), 21
plot.poisson_greenreg (reg_poisson), 24
plot.rlm_greenreg (rlm), 26
plot.rls_greenreg (rls), 28
print.ar_greenreg (modelo_ar), 11
print.arima_greenreg (modelo_arima), 14
print.arma_greenreg (modelo_arma), 18
print.logistico_greenreg (logistico), 9
print.ma_greenreg (modelo_ma), 21
print.poisson_greenreg (reg_poisson), 24
print.rlm_greenreg (rlm), 26
print.rls_greenreg (rls), 28

reg_poisson, 24

rlm, 26
rls, 28