Package: GREENREG (via r-universe)

October 19, 2026

Title Herramienta para el Análisis Estadístico y Ambiental

Version 0.1.0

Description Provee un conjunto de funciones accesibles y automatizadas para aplicar modelos estadísticos (RLS, RLM, GLM) y series de tiempo (ARIMA). Diseñado para apoyar la enseñanza en la Universidad Autónoma Chapingo, facilitando la interpretación de resultados y la validación de supuestos mediante diagnósticos gráficos automáticos.

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 3.5)

Imports ggplot2, ggrepel, stats, graphics, grDevices, lmtest,

Suggests pROC, broom, car, rmarkdown, tseries, testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://dbazan2003.r-universe.dev

Date/Publication 2026-04-17 03:46:01 UTC

RemoteUrl https://github.com/dbazan2003/greenreg

RemoteRef HEAD

RemoteSha 281c452ecbb1c8a787eaa47173ca58f416bbaa7e

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 10
modelo_arima 12
modelo_arma 13
modelo_ma 15
plot.ar_greenreg 16
plot.arma_greenreg 17
plot.logistico_greenreg 17
plot.ma_greenreg 17
plot.poisson_greenreg 18
plot.rlm_greenreg 18
plot.rls_greenreg 18
print.ar_greenreg 19
print.arima_greenreg 19
print.arma_greenreg 19
print.logistico_greenreg 20
print.ma_greenreg 20
print.poisson_greenreg 20
print.rlm_greenreg 21
print.rls_greenreg 21
reg_poisson 22
rlm 23
rls 25

Index **28**

`analisis_datos` *Asistente Interactivo de Análisis de Datos*

Description

Esta es la función principal y punto de entrada del paquete. Actúa como un consultor automatizado que guía al usuario desde la carga de datos hasta la generación del código final, seleccionando la metodología adecuada según la naturaleza de la variable objetivo.

Usage

```
analisis_datos(data, variable_objetivo)
```

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").

Value

No devuelve un objeto al entorno (return invisible). Su función es:

1. Imprimir en consola el **código R listo para copiar y pegar**.
2. Generar el gráfico exploratorio en el panel de gráficos.

¿Cómo funciona este Asistente paso a paso?

1. **Escaneo y Validación:** Analiza la base de datos para asegurar que la variable objetivo existe. Además, realiza un test interno de autocorrelación para verificar si tiene sentido aplicar modelos de Series de Tiempo (evitando análisis espurios).
2. **Menú Interactivo:** Despliega opciones en la consola basándose en los datos detectados, preguntando al usuario qué enfoque desea (Regresión Clásica, Modelos Generalizados o Series de Tiempo).
3. **Generación de Estrategias (El "Motor"):**
   - *Para Regresión y GLM:* Crea dos versiones del código.
     (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 "Búsqueda en Rejilla" (Grid Search) probando combinaciones de parámetros (p, d, q). Identifica automáticamente si la serie necesita diferenciación (ARIMA), si es puramente autoregresiva (AR), de media móvil (MA) o mixta (ARMA), y genera el código para la función específica.

Guía de la Gráfica Exploratoria Generada

Dependiendo de la opción elegida, la función genera una visualización preliminar diferente para entender los datos antes de modelar:

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

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

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

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

Examples

```
# Supongamos que tienes una base llamada 'rendimiento_maiz_ton_ha'
# maiz<- analisis_datos(datos_rendimiento_maiz, "rendimiento_maiz_ton_ha")
# 1. Se abrirá el menú.
```

```
# 2. Eliges opción 1 (Regresión).
# 3. Copias el código que aparece en consola para ajustar tu modelo.
```

`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)

anomalia_c Anomalía de temperatura respecto al promedio histórico (°C)

Details

Los datos fueron construidos de manera simulada a partir de patrones reales observados en bases climáticas del INEGI y de la CONAGUA. La creación propia de esta base permite garantizar que los valores sean adecuados para fines didácticos y compatibles con las funciones del paquete **CHAPIREG**.

Source

Elaboración propia con fines didácticos a partir de información pública de INEGI y CONAGUA.

Examples

```
# Probar un modelo Autorregresivo de orden 2
modelo_ar(anomalia_temperatura$anomalia_c, p = 2)

# Probar un modelo ARIMA(1,1,1)
modelo_arima(anomalia_temperatura$anomalia_c, p = 1, d = 1, q = 1)
```

`datos_nivel_presa` *Niveles de agua en presas de México*

Description

Conjunto de datos que presenta los niveles de almacenamiento de agua registrados en distintas presas y fechas, útil para análisis 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 didácticos para demostrar la aplicación de modelos AR, MA y ARIMA en **CHAPIREG**.

Source

Elaboración propia con fines didácticos basada en patrones de datos de CONAGUA.

Examples

```
# Visualizar la serie
plot(nivel_presa$fecha, nivel_presa$nivel_m, type = "l")

# Ajustar un modelo ARIMA(1,1,1)
modelo_complejo <- modelo_arima(nivel_presa$nivel_m, p = 1, d = 1, q = 1)
```

`datos_rendimiento_maiz` *Rendimiento de maíz bajo distintos niveles de fertilización*

Description

Conjunto de datos que presenta el rendimiento del maíz en función de diferentes variables climáticas y de manejo del cultivo. Este dataset es útil para ejemplificar modelos de regresión lineal simple y múltiple.

Usage

```
datos_rendimiento_maiz
```

Format

Un data frame con 250 observaciones y 4 variables:

rendimiento_maiz_ton_ha Rendimiento del cultivo de maíz (ton/ha)

precipitacion_mm Precipitación registrada durante el ciclo agrícola (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 diseñó específicamente para probar las funciones de regresión del paquete **CHAPIREG**.

Source

Elaboración propia con fines didácticos tomando referencia de datos del SIAP e INEGI.

Examples

```
# Análisis completo con regresión múltiple
modelo_completo <- rlm(rendimiento_maiz_ton_ha ~ precipitacion_mm + temperatura_c + fertilizante_kg_ha,
                       data = rendimiento_maiz)

# Análisis simple para ver solo el efecto de la lluvia
modelo_simple <- rls(rendimiento_maiz_ton_ha ~ precipitacion_mm, data = rendimiento_maiz)

# Aplicar Análisis de Componentes Principales (ACP)
predictores <- rendimiento_maiz[, c("precipitacion_mm", "temperatura_c", "fertilizante_kg_ha")]
modelo_acp(predictores, escala = TRUE)
```

`datos_roya_cafe` *Presencia de roya del café en diferentes localidades*

Description

Base de datos simulada que presenta la presencia/ausencia de roya del café junto con variables ambientales y de manejo agronómico relevantes. Es útil para ejercicios de regresión logística.

Usage

```
datos_roya_cafe
```

Format

Un data frame con 300 observaciones y 4 variables:

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

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 plantación

Details

Los datos fueron generados de forma controlada utilizando parámetros similares a los observados por SIAP y estudios agronómicos del INIFAP. Esta base fue creada para ser utilizada en funciones de regresión logística del paquete **CHAPIREG**.

Source

Elaboración propia con fines didácticos basada en información de SIAP e INIFAP.

Examples

```
# Predecir la presencia o ausencia de roya
modelo_roya <- logistico(presencia_roya ~ humedad_relativa_pct + altitud_msnm + manejo_sombra,
                         data = roya_cafe)
```

`datos_tala_ilegal_poisson` *Registros de tala ilegal en distintas regiones*

Description

Conjunto de datos que contiene el número 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 aplicación de modelos de regresión Poisson.

Usage

```
datos_tala_ilegal_poisson
```

Format

Un data frame con 200 observaciones y 4 variables:

numero_talas Número de eventos de tala ilegal observados

densidad_poblacion Densidad poblacional de la región (hab/km²)

superficie_forestal_ha Superficie forestal en hectáreas

dias_vigilancia Número de días de vigilancia reportados

Details

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.

Source

Elaboración propia con fines didácticos tomando referencia de CONAFOR e INEGI.

Examples

```
# Modelar el número de talas ilegales
modelo_conteo <- poisson(numero_talas ~ densidad_poblacion + superficie_forestal_ha + dias_vigilancia,
                         data = datos_tala_ilegal_poisson)
```

`logistico` *Regresión Logística Binaria (MLG Logit)*

Description

Esta función ajusta un Modelo Lineal Generalizado (GLM) de la familia binomial para predecir la probabilidad de un evento dicotómico (Sí/No, 0/1). A diferencia de la regresión lineal, el motor de cálculo estima los coeficientes mediante Máxima Verosimilitud (MLE) y ofrece un diagnóstico especializado en clasificación.

Usage

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

Arguments

`formula`	Una fórmula 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).

Details

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

1. **Ajuste Logit:** Transforma la variable respuesta mediante la función de enlace $logit(p) = ln(p/(1-p))$, permitiendo que la probabilidad se mantenga entre 0 y 1.
2. **Conversión de Impacto:** Los coeficientes estimados se transforman automáticamente a **Odds Ratios (OR)** mediante e^{β}. Esto permite concluir cuánto multiplica cada variable la posibilidad de que ocurra el evento.
3. **Evaluación de Bondad (McFadden):** Calcula el Pseudo-R^2 comparando la verosimilitud del modelo ajustado contra un modelo nulo (sin predictores).
4. **Métricas de Clasificación:** Genera la Matriz de Confusión basada en el `umbral` especificado y calcula la Precisión Global (Accuracy).
5. **Diagnóstico de Influencia:** Evalúa si existen observaciones que deforman las probabilidades mediante el cálculo de la Distancia de Cook y el Apalancamiento (Leverage).

Value

Un objeto de clase `"logistico_greenreg"` que contiene:

- `modelo`: Objeto `glm` original.
- `odds_ratios`: Tabla con OR, intervalos de confianza y p-valores.
- `metricas`: Lista con Pseudo-R2, AIC y Accuracy.
- `confusion`: Tabla cruzada de Real vs. Predicho.
- `predicciones`: Dataframe con probabilidades y métricas de diagnóstico (leverage/cook).
- `notas`: Reporte interpretado con semáforos de validación técnica.

Guía de las 9 Gráficas de Diagnóstico (`plot`)

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

1. **Capacidad de Separación** Muestra si el modelo realmente distingue entre grupos. Buscamos "montañas" de densidad que no se traslapen.
2. **Matriz de Confusión 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 Calibración** 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 diagnóstico. Un AUC de 1.0 es una clasificación perfecta; 0.5 es puro azar.
6. **Distancia de Cook** Detecta casos atípicos en la clasificación que tienen un peso excesivo en los coeficientes.
7. **Trade-off de Umbrales** Permite ver cómo cambia la Sensibilidad y Especificidad al mover el punto de corte.
8. **Residuos de Pearson** Busca patrones sistemáticos en el error. Idealmente, la línea 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 = da
# modelo_log
# plot(modelo_log)
```

`modelo_ar` *Modelo Autorregresivo (AR)*

Description

Esta función ajusta un modelo de series de tiempo donde el valor actual (Y_t) se explica mediante una combinación lineal de sus propios valores pasados ($Y_t-1, Y_t-2, ...$). Es la herramienta estándar para analizar la "persistencia" o "memoria" de un fenómeno a través del tiempo.

Usage

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

Arguments

`x`	Vector numérico o serie de tiempo (`ts`). No debe contener valores `NA`.
`p`	Orden del modelo (número de rezagos a considerar). Por defecto es 1.
`nivel_confianza`	Nivel de significancia para las bandas de error (default 0.95).

Details

El flujo interno de la función se divide en cinco etapas diagnósticas:

1. **Estimación (MLE):** Utiliza Máxima Verosimilitud para encontrar los coeficientes ϕ (Phi) que mejor describen la dependencia temporal.
2. **Análisis de Estabilidad:** Calcula las raíces inversas del polinomio característico. Un modelo AR solo es válido si es "Estacionario", es decir, si las raíces se encuentran dentro del círculo unitario.
3. **Inferencia de Parámetros:** Calcula errores estándar y p-valores para determinar cuántos rezagos (lags) son realmente significativos.
4. **Auditoría de Residuos:** Evalúa si el error restante es "Ruido Blanco" mediante la prueba de Ljung-Box (independencia) y Shapiro-Wilk (normalidad).
5. **Traducción Didáctica:** 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 los coeficientes, las pruebas de estabilidad, los diagnósticos de ruido blanco y dataframes listos para visualización avanzada.

Guía de las 9 Gráficas de Diagnóstico (`plot`)

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

1. **Inspección Visual** Muestra la serie bruta con su media histórica. Ayuda a detectar tendencias que el modelo AR no puede manejar por sí 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. **Círculo Unitario** Gráfica matemática crítica. Si los rombos rojos salen del círculo, el modelo es inestable y no sirve para pronosticar.
5. **ACF de Residuos** Busca correlaciones remanentes. Las barras deben ser pequeñas 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 distribución normal, requisito para que los intervalos de confianza sean exactos.

8. **Precisión (Real vs Ajustado)** Muestra la correlación directa entre la predicción y el dato observado.
9. **Estacionariedad Móvil** Analiza si la media y la varianza cambian a lo largo de la serie. Una línea 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)
```

---

modelo_arima *Modelo Integrado de Media Móvil Autorregresiva (ARIMA)*

Description

Esta función ajusta el modelo más versátil para series de tiempo univariadas: el ARIMA(p, d, q). Combina la memoria de largo plazo (AR), la estabilización mediante diferenciación (I) para eliminar tendencias, y la respuesta a choques aleatorios de corto plazo (MA).

Usage

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

Arguments

`x`	Vector numérico o serie de tiempo (`ts`).
`p`	Orden Autorregresivo (memoria del pasado).
`d`	Orden de Integración (número de diferenciaciones).
`q`	Orden de Media Móvil (memoria de los errores).
`include_mean`	Lógico. ¿Incluir constante? (Suele ser `FALSE` si $d > 0$).

Details

El flujo interno de la función ejecuta una auditoría de cinco fases:

1. **Diferenciación (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. **Estimación ARMA:** Sobre la serie ya estabilizada, estima los parámetros ϕ (AR) y θ (MA) mediante Máxima Verosimilitud.
3. **Doble Validación de Estructura:** Analiza las raíces del polinomio característico para asegurar que la parte AR sea **Estable** y la parte MA sea **Invertible**.
4. **Auditoría de Residuos:** Verifica que el error final sea "Ruido Blanco" mediante las pruebas de Ljung-Box (independencia) y Shapiro-Wilk (normalidad).
5. **Diagnóstico Dinámico:** Evalúa si la media y la varianza se mantienen estables a lo largo del tiempo después de la integración.

Value

Un objeto de clase `"arima_greenreg"` que contiene el modelo `stats::arima`, tabla de coeficientes, estatus de estabilidad, serie diferenciada y dataframes para visualización avanzada.

Guía de las 10 Gráficas de Diagnóstico (`plot`)

Al ejecutar `plot(modelo)`, se despliega el diagnóstico más 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\ \item**3. Serie Diferenciada**Muestra la serie después de aplicar d. Si el modelo es correcto, esta serie debe verse plana y sin tendencia. \item**4. Círculo Unitario**Gráfica de convergencia matemática. Los puntos rojos (AR) y azules (MA) deben estar dentro del círculo gris. \item**5. Residuos en el Tiempo**Busca patrones de error. Los puntos deben formar una nube aleatoria sin ciclos visibles. \item**6. ACF de Residuos**Prueba de independencia. Las barras deben estar dentro de los límites para confirmar el Ruido Blanco. \item**7. PACF de Residuos**Ayuda a identificar si el orden autorregresivo p es suficiente para explicar la inercia. \item**8. Normalidad (Q-Q Plot)**Valida que los errores sigan una distribución gaussiana para asegurar pronósticos confiables. \item**9. Precisión (Real vs Ajustado)**Evaluación de la correlación de Pearson entre los datos y la estimación. \item**10. Estacionariedad Final**Muestra la media y varianza móvil. Una franja púrpura estable certifica un modelo robusto.

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

arima

`modelo_arma` *Modelo Autorregresivo de Media Móvil (ARMA)*

Description

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

Usage

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

Arguments

`x`	Vector numérico o serie de tiempo (`ts`).
`p`	Orden Autorregresivo (pasados de la variable).
`q`	Orden de Media Móvil (pasados del error).
`include_mean`	Lógico. ¿Incluir constante/intercepto? Por defecto `TRUE`.

Details

El flujo interno de la función realiza una auditoría estructural en cinco pasos:

1. **Estimación Conjunta:** Utiliza Máxima Verosimilitud (MLE) para encontrar simultáneamente los coeficientes ϕ (AR) y θ (MA).
2. **Estabilidad (Parte AR):** Verifica que el componente autorregresivo no sea explosivo. Las raíces inversas del polinomio AR deben estar dentro del círculo.
3. **Invertibilidad (Parte MA):** Verifica que el componente de media móvil sea unívoco. Las raíces inversas del polinomio MA deben estar dentro del círculo.
4. **Diagnóstico de Residuos:** Evalúa si el error resultante es Ruido Blanco mediante el test de Ljung-Box, asegurando que no quede señal sin capturar.
5. **Criterio de Información:** Proporciona el AIC para facilitar la comparación entre diferentes combinaciones de (p, q), buscando el equilibrio entre complejidad y precisión.

Value

Un objeto de clase `"arma_greenreg"` que contiene los coeficientes mixtos, el estatus de estabilidad/invertibilidad, diagnósticos de residuos y coordenadas para el círculo unitario doble.

Guía de las 9 Gráficas de Diagnóstico (`plot`)

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

1. **Inspección 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 cómo interactúan la inercia y los shocks.
3. **Residuos en el Tiempo** Evaluación de la aleatoriedad del error. Los puntos deben formar una nube sin patrones.
4. **Círculo Unitario Dual** Gráfica crítica exclusiva de ARMA. Muestra simultáneamente la Estabilidad (Puntos Rojos) e Invertibilidad (Rombos Azules).
5. **ACF de Residuos** Busca autocorrelaciones remanentes. Las barras deben ser mínimas para confirmar que el modelo es óptimo.
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. **Precisión (Real vs Ajustado)** Muestra la correlación directa. Un ARMA exitoso suele tener una alineación estrecha sobre la diagonal.
9. **Estacionariedad Móvil** 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)
```

`modelo_ma` *Modelo de Media Móvil (MA)*

Description

Esta función ajusta un modelo de series de tiempo donde el valor actual (Y_t) se explica mediante una combinación lineal de los errores (shocks) pasados ($\epsilon_t-1, \epsilon_t-2, ...$). Es ideal para modelar fenómenos con efectos de memoria corta o impactos transitorios.

Usage

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

Arguments

`x`	Vector numérico o serie de tiempo (`ts`). No debe contener valores `NA`.
`q`	Orden del modelo (número de choques pasados a considerar). Por defecto es 1.
`include_mean`	Lógico. ¿Incluir el promedio constante de la serie? Por defecto `TRUE`.

Details

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

1. **Estimación (MLE):** Utiliza Máxima Verosimilitud para estimar los parámetros 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 raíces inversas del polinomio característico. Un modelo MA debe ser "Invertible" para que el error pueda expresarse como una función convergente de los datos observados.
3. **Identificación Estructural:** Evalúa la ACF (Autocorrelación). En un proceso MA(q), la ACF debe "cortarse" abruptamente después del lag q.
4. **Auditoría de Residuos:** Verifica que el error remanente sea Ruido Blanco mediante la prueba de Ljung-Box y la distribución Gaussiana (Shapiro-Wilk).
5. **Traducción Didáctica:** Genera un reporte que explica la persistencia de los choques y valida si el modelo es único y matemáticamente sólido.

Value

Un objeto de clase "ma_greenreg" con coeficientes θ, pruebas de invertibilidad, diagnósticos de residuos y dataframes para visualización con ggplot2.

Guía de las 9 Gráficas de Diagnóstico (plot)

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

1. **Inspección Visual** Muestra la serie bruta con su media histórica. El modelo MA supone que la serie siempre es atraída hacia este nivel central.
2. **Ajuste Temporal** Superpone el modelo (azul) sobre la realidad. Permite ver cómo el modelo "suaviza" los picos de la serie.
3. **Precisión (Real vs Ajustado)** Gráfica de dispersión para evaluar la correlación. Los puntos deben seguir la diagonal roja.
4. **Círculo Unitario** Prueba de Invertibilidad. Los rombos azules deben estar dentro del círculo para asegurar que el modelo sea único.
5. **Residuos en el Tiempo** Evaluación de aleatoriedad. No debe haber patrones; los errores deben ser impredecibles.
6. **ACF de Residuos** Prueba crítica de Ruido Blanco. Si hay barras largas (números 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 Móvil** 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)
```

---

plot.ar_greenreg *Gráficas para Modelo AR*

Description

Gráficas para Modelo AR

Usage

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

`plot.arma_greenreg` *Gráficas para Modelo ARMA*

Description

Gráficas para Modelo ARMA

Usage

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

`plot.logistico_greenreg` *Generación de Gráficas para Regresión Logística*

Description

Generación de Gráficas para Regresión Logística

Usage

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

`plot.ma_greenreg` *Gráficas Didácticas para Modelo MA*

Description

Gráficas Didácticas para Modelo MA

Usage

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

`plot.poisson_greenreg` *Generación de Gráficas*

Description

Generación de Gráficas

Usage

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

`plot.rlm_greenreg` *Gráficas para RLM*

Description

Gráficas para RLM

Usage

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

`plot.rls_greenreg` *Gráficas de diagnóstico para rls_greenreg*

Description

Genera una secuencia de 8 gráficas para validar el modelo.

Usage

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

Arguments

`x`	Objeto de clase rls_greenreg.
`...`	Argumentos adicionales.

print.ar_greenreg *Impresión para Modelos Autorregresivos (AR)*

Description

Impresión para Modelos Autorregresivos (AR)

Usage

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

print.arima_greenreg *Impresión Académica para Modelos Integrados ARIMA(p, d, q)*

Description

Impresión Académica para Modelos Integrados ARIMA(p, d, q)

Usage

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

print.arma_greenreg *Impresión para Modelo ARMA*

Description

Impresión para Modelo ARMA

Usage

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

`print.logistico_greenreg` *Impresión para Regresión Logística*

Description

Impresión para Regresión Logística

Usage

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

`print.ma_greenreg` *Impresión Académica para Modelos de Media Móvil (MA)*

Description

Impresión Académica para Modelos de Media Móvil (MA)

Usage

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

`print.poisson_greenreg` *Impresión Académica para Regresión de Poisson*

Description

Impresión Académica para Regresión de Poisson

Usage

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

print.rlm_greenreg *Reporte de Regresión Lineal Múltiple*

Description

Imprime un resumen detallado del modelo RLM.

Usage

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

Arguments

x	Objeto de clase rlm_greenreg.
...	Argumentos adicionales.

print.rls_greenreg *Impresión de resultados rls_greenreg*

Description

Método optimizado para imprimir el reporte de la regresión.

Usage

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

Arguments

x	Objeto de clase rls_greenreg.
...	Argumentos adicionales.

`reg_poisson` *Regresión de Poisson (Modelos de Conteo)*

Description

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

Usage

```
reg_poisson(formula, data)
```

Arguments

`formula`	Una fórmula de R (ej. `numero_frutos ~ riego + fertilizante`).
`data`	Un `data.frame` con las observaciones (el conteo debe ser entero).

Details

El flujo interno de la función se divide en seis etapas críticas:

1. **Ajuste Log-Lineal:** Estima la relación entre los predictores y el logaritmo de la tasa de eventos (λ). Utiliza Máxima Verosimilitud (MLE) para el ajuste.
2. **Interpretación de Tasas (IRR):** Los coeficientes originales (β) se exponencian automáticamente (e^β) para obtener los **Ratios de Tasas de Incidencia**. Esto permite concluir en qué porcentaje aumenta o disminuye la frecuencia de eventos por cada unidad de cambio en X.
3. **Diagnóstico de Equidispersión:** Calcula el Ratio de Dispersión basado en el estadístico de Pearson (X^2 / gl). En una distribución Poisson perfecta, este valor debe ser 1. Valores > 1.5 indican **Sobredispersión**, sugiriendo que los datos varían más 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. **Evaluación de Residuos:** Calcula residuos de devianza y Pearson para identificar patrones de error sistemáticos o falta de ajuste en conteos específicos (como el cero).
6. **Traducción Didáctica:** Genera un reporte que advierte si el modelo es válido o si se debe migrar a un modelo Binomial Negativa debido a la dispersión.

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 dispersión calculado (Pearson / GL).
- `metricas`: Lista con AIC, Log-Verosimilitud y Devianzas.
- `predicciones`: Dataframe con valores observados, lambda estimado y diagnósticos (leverage/residuos).

Guía de las 8 Gráficas de Diagnóstico (`plot`)

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

1. **Distribución de Conteos (Real vs Modelo)** Compara las frecuencias observadas contra las que el modelo "esperaba" predecir. Crucial para detectar el exceso de ceros.
2. **Precisión (Predicho vs Real)** Scatterplot entre el Lambda estimado y el conteo real. Los puntos deben seguir la tendencia de la línea 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_vigila
# modelo_poisson
# plot(modelo_poisson)
```

`rlm` *Regresión Lineal Múltiple (RLM)*

Description

Esta función ajusta un modelo de regresión lineal con múltiples variables predictoras ($y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...$) 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.

Usage

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

Arguments

`formula`	Fórmula modelo.
`data`	Dataframe con los datos.
`nivel_confianza`	Nivel de confianza para intervalos (default 0.95).

Details

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

1. **Validación de Datos:** Asegura que la fórmula sea válida y que el dataframe no contenga errores estructurales.
2. **Ajuste (Engine):** Utiliza `stats::lm`. En RLM, el foco principal es el **R^2 Ajustado**, que penaliza la inclusión de variables innecesarias que no aportan información real.
3. **Detección de Multicolinealidad (VIF):** Calcula manualmente el Factor de Inflación de la Varianza ($VIF = 1 / (1 - R^2_j)$) para cada predictor. Esto identifica si las variables X están correlacionadas entre sí, lo cual infla los errores estándar y vuelve inestables los coeficientes.
4. **Auditoría de Supuestos:** Ejecuta de forma nativa pruebas de Normalidad (Shapiro), Homocedasticidad (Breusch-Pagan), Independencia (Durbin-Watson) y Linealidad (Rainbow Test).
5. **Traducción Automática:** Evalúa los estadísticos y genera "Notas" (, ,) y sugerencias de mejora (ej. transformar variables o eliminar predictores redundantes).
6. **Preparación de Diagnóstico:** 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.
- `vif`: Vector con los valores de inflación de varianza por variable.
- `supuestos`: Resultados de los tests de Shapiro, BP, DW y Rainbow.
- `notas`: Reporte interpretado con semáforos de advertencia.
- `data_plot`: Dataframe listo para visualización avanzada.

Guía de las 10 Gráficas de Diagnóstico (`plot`)

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

1. **Distribución de Variables (Y, Xi)** Muestra histogramas y densidades de todas las variables. Permite detectar sesgos o datos atípicos antes del ajuste.
2. **Relación 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 precisión global. Los puntos deben seguir la diagonal de 45°.
4. **Q-Q Plot (Normalidad)** Valida si los residuos siguen una distribución 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 recolección. Detecta patrones temporales o espaciales en los errores.
7. **Predicción vs Realidad (Identidad)** Versión simplificada del ajuste para evaluar la correlación de Pearson.
8. **Multicolinealidad (VIF)** Gráfica 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 actúan como palancas sobre la recta.

Examples

```
# data(datos_rendimiento_maiz)
# modelo_rlm <- rlm(rendimiento_maiz_ton_ha ~., data = datos_rendimiento_maiz)
# modelo_rlm       # Ver reporte
# plot(modelo_rlm) # Ver gráficas
```

`rls` *Regresión Lineal Simple (RLS)*

Description

Esta función ajusta un modelo de regresión lineal ($y = \beta_0 + \beta_1 x$) mediante Mínimos Cuadrados Ordinarios (OLS). A diferencia de `lm()` estándar, esta función devuelve un objeto enriquecido con interpretaciones en lenguaje natural, verificaciones automáticas de supuestos y sugerencias para el investigador.

Usage

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

Arguments

`formula`	Una fórmula de R estándar (ej. `rendimiento ~ fertilizante`).
`data`	El data.frame que contiene las variables.
`nivel_confianza`	Nivel de confianza para los intervalos (por defecto 0.95).

Details

El flujo interno de la función se divide en cinco etapas críticas:

1. **Preparación:** Valida que los datos sean coherentes y extrae las variables de la fórmula.
2. **Estimación:** Calcula los coeficientes, el error estándar residual (σ) y la significancia global (F-test).
3. **Inferencia:** Genera intervalos de confianza para los parámetros (β) al nivel especificado.
4. **Auditoría de Supuestos:** Ejecuta pruebas de Shapiro-Wilk (Normalidad), Breusch-Pagan (Homocedasticidad) y Durbin-Watson (Independencia).
5. **Traducción:** Convierte los estadísticos abstractos en "Notas" de texto que recomiendan acciones específicas (ej. aplicar logaritmos o revisar datos atípicos).

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 y P-valores.
- `supuestos`: Resultados de los tests de Shapiro, BP y DW.
- `notas`: Diccionario de interpretaciones y sugerencias de mejora.
- `data_plot`: Dataframe listo para ggplot2 con residuos estandarizados e índices de influencia.

Guía de las 9 Gráficas de Diagnóstico (`plot`)

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

1. **Histogramas Exploratorios** Compara la distribución de X e Y. Útil para detectar sesgos iniciales en las variables originales.
2. **Relación Original (Scatterplot)** Muestra la nube de puntos pura. Ayuda a confirmar visualmente si la relación parece lineal o curva antes de ver la recta.
3. **Recta de Regresión** Visualiza el ajuste final con bandas de confianza. Permite ver qué tan bien la recta representa la tendencia de los datos.
4. **Q-Q Plot (Normalidad)** Grafica los cuantiles de los residuos contra una normal teórica. Si los puntos se alejan de la diagonal, los p-valores del modelo podrían no ser válidos.
5. **Homocedasticidad** Residuos vs. Predichos. Si la dispersión aumenta (forma de abanico), el modelo pierde precisión en valores altos.

6. **Independencia** Residuos vs. Orden de datos. Vital para detectar si el error de hoy depende del error de ayer (Autocorrelación).
7. **Precisión (Predicho vs Real)** Muestra la correlación directa. En un modelo perfecto, todos los puntos caerían sobre la línea de identidad (45°).
8. **Distancia de Cook** Identifica observaciones "influyentes". Una barra alta indica un dato que, si se elimina, cambiaría drásticamente la pendiente del modelo.
9. **Leverage vs Residuos** Diferencia entre 'Outliers' (error en Y) y 'Apalancamiento' (valor extremo en X). Identifica puntos que "jalonean" la recta injustificadamente.

Examples

```
# data(maiz)
# modelo <- rls(rendimiento_maiz_ton_ha ~ precipitacion_mm, data = datos_rendimiento_maiz)
# modelo       # Imprime el reporte
# plot(modelo) # Genera las 9 gráficas
```

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, *13*

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, 10
modelo_arima, 12
modelo_arma, 13
modelo_ma, 15

plot.ar_greenreg, 16
plot.arma_greenreg, 17
plot.logistico_greenreg, 17
plot.ma_greenreg, 17
plot.poisson_greenreg, 18
plot.rlm_greenreg, 18
plot.rls_greenreg, 18
print.ar_greenreg, 19
print.arima_greenreg, 19
print.arma_greenreg, 19
print.logistico_greenreg, 20
print.ma_greenreg, 20
print.poisson_greenreg, 20
print.rlm_greenreg, 21
print.rls_greenreg, 21

reg_poisson, 22
rlm, 23
rls, 25