Package ‘assessor’

February 16, 2025

Title Assessment Tools for Regression Models with Discrete and Semicontinuous Outcomes

Version 1.1.1

Description Provides assessment tools for regression models with discrete and semicontinuous outcomes proposed in Yang (2023) <doi:10.48550/arXiv.2308.15596>. It calculates the double probability integral transform (DPIT) residuals, constructs QQ plots of residuals and the ordered curve for assessing mean structures.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

URL `https://jhlee1408.github.io/assessor/`

BugReports `https://github.com/jhlee1408/assessor/issues`

Imports tweedie, MASS, VGAM, np

Suggests pscl, statmod, rmarkdown, knitr, AER, faraway, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Lu Yang [aut],
Jeonghwan Lee [cre, aut]

Maintainer Jeonghwan Lee <lee03938@umn.edu>

Contents

`bballHR` *MLB Players' Home Run and Batted Ball Statistics with Red Zone Metrics (2017-2019)*

Description

This dataset provides annual statistics for Major League Baseball (MLB) players, including home run counts, at-bats, mean exit velocities, launch angles, quantile statistics of exit velocities and launch angles, and red zone metrics. It is intended for analyzing batted ball performance, with additional variables on the red zone, which is defined as balls in play with a launch angle between 20 and 35 degrees and an exit velocity of at least 95 mph.

Usage

```
bballHR
```

Format

A data frame with the following columns:

name Player's full name (character).

playerID Player's unique identifier in the Lahman database (character).

teamID Team abbreviation (character).

year Season year (numeric).

HR Home runs hit during the season (integer).

AB At-bats during the season (integer).

mean_exit_velo Average exit velocity (mph) over the season (numeric).

mean_launch_angle Average launch angle (degrees) over the season (numeric).

launch_angle_75 Launch angle at the 75th percentile of the player's distribution (numeric).

launch_angle_70 Launch angle at the 70th percentile of the player's distribution (numeric).

launch_angle_65 Launch angle at the 65th percentile of the player's distribution (numeric).

exit_velo_75 Exit velocity at the 75th percentile of the player's distribution (numeric).

exit_velo_80 Exit velocity at the 80th percentile of the player's distribution (numeric).

exit_velo_85 Exit velocity at the 85th percentile of the player's distribution (numeric).

count_red_zone Seasonal count of batted balls in the red zone, defined as a launch angle between 20 and 35 degrees and an exit velocity greater than or equal to 95 mph (integer).

prop_red_zone Proportion of batted balls that fall into the red zone (numeric).

BPF Ballpark factor, indicating the effect of the player's home ballpark on offensive statistics (integer).

Details

Mean Metrics `mean_exit_velo` and `mean_launch_angle` represent the player's average exit velocities and launch angles, respectively, over the course of a season.

Quantile Metrics The `launch_angle_xx` and `exit_velo_xx` columns denote the upper x-percentiles (e.g., 75th percentile) of the player's launch angle and exit velocity distributions for that year.

Red Zone Metrics `count_red_zone` gives the number of balls in play that fall into the red zone, while `prop_red_zone` represents the proportion of balls in play in this category.

BPF The Ballpark Factor (BPF) quantifies the influence of the player's home ballpark on offensive performance, with values above 100 indicating a hitter-friendly environment.

Source

- Player statistics: Lahman R Package
- Batted ball data: Baseball Savant
- Additional analysis: Patterns of Home Run Hitting in the Statcast Era by Jim Albert

Examples

```
data(bballHR)
head(bballHR)
```

`MEPS` *Healthcare expenditure data*

Description

Healthcare expenditure data set.

Usage

```
MEPS
```

Format

A data frame with 29784 rows and 29 variables:

`EXP` the aggregate annual office based expenditure per participants, semicontinuous outcomes

`AGE` Age

`GENDER` 1 if female

`ASIAN` 1 if Asian

`BLACK` 1 if Black

`NORTHEAST` 1 if Northeast

`MIDWEST` 1 if Midwest

`SOUTH` 1 if South

`USC` 1 if have usual source of care

`COLLEGE` 1 if colleage or higher degrees

HIGHSCH 1 if high school degree

MARRIED 1 if married

WIDIVSEP 1 if widowed or divorced or separated

FAMSIZE Family Size

HINCOME 1 if high income

MINCOME 1 if middle income

LINCOME 1 if low income

NPOOR 1 if near poor

POOR 1 if poor

FAIR 1 if fair

GOOD 1 if good

VGOOD 1 if very good

MNHPOOR 1 if poor or fair mental health

ANYLIMIT 1 if any functional or activity limitation

unemployed 1 if unemployed at the beginning of 2006

EDUCHEALTH 1 if education, health and social services

PUBADMIN 1 if public administration

insured 1 if is insured at the beginning of the year 2006

MANAGEDCARE if enrolled in an HMO or a gatekeeper plan

Source

http://www.meps.ahrq.gov/mepsweb/

`ord_curve` *Ordered curve for assessing mean structures*

Description

Creates a plot to assess the mean structure of regression models. The plot compares the cumulative sum of the response variable and its hypothesized value. Deviation from the diagonal suggests the possibility that the mean structure of the model is incorrect.

Usage

```
ord_curve(model, thr)
```

Arguments

`model`	Regression model object (e.g.,`lm`, `glm`, `glm.nb`, `polr`, `lm`)
`thr`	Threshold variable (e.g., predictor, fitted values, or variable to be included as a covariate)

Details

The ordered curve plots

$$\hat{L}_1(t) = \frac{\sum_{i=1}^{n}\left[Y_i 1(Z_i \le t)\right]}{\sum_{i=1}^{n} Y_i}$$

against

$$\hat{L}_2(t) = \frac{\sum_{i=1}^{n}\left[\hat{\lambda}_i 1(Z_i \le t)\right]}{\sum_{i=1}^{n} \hat{\lambda}_i},$$

where $\hat{\lambda}_i$ is the fitted mean, and Z_i is the threshold variable.

If the mean structure is correctly specified in the model, $\hat{L}_1(t)$ and $\hat{L}_2(t)$ should be close to each other.

If the curve is distant from the diagonal, it suggests incorrectness in the mean structure. Moreover, if the curve is above the diagonal, the summation of the response is larger than the fitted mean, which implies that the mean is underestimated, and vice versa.

The role of `thr` (threshold variable Z) is to determine the rule for accumulating $\hat{\lambda}_i$ and Y_i, $i = 1, \ldots, n$ for the ordered curve. The candidate for `thr` could be any function of predictors such as a single predictor (e.g., `x1`), a linear combination of predictor (e.g., `x1+x2`), or fitted values (e.g., `fitted(model)`). It can also be a variable being considered to be included in the mean function. If a variable leads to a large discrepancy between the ordered curve and the diagonal, including this variable in the mean function should be considered.

For more details, see the reference paper.

Value

- x-axis: $\hat{L}_1(t)$
- y-axis: $\hat{L}_2(t)$

which are defined in Details.

References

Yang, Lu. "Double Probability Integral Transform Residuals for Regression Models with Discrete Outcomes." arXiv preprint arXiv:2308.15596 (2023).

Examples

```
## Binary example of ordered curve
n <- 500
set.seed(1234)
x1 <- rnorm(n, 1, 1)
x2 <- rbinom(n, 1, 0.7)
beta0 <- -5
beta1 <- 2
beta2 <- 1
beta3 <- 3
q1 <- 1 / (1 + exp(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x1 * x2))
y1 <- rbinom(n, size = 1, prob = 1 - q1)

## True Model
model0 <- glm(y1 ~ x1 * x2, family = binomial(link = "logit"))
```

```
ord_curve(model0, thr = model0$fitted.values) # set the threshold as fitted values

## Missing a covariate
model1 <- glm(y1 ~ x1, family = binomial(link = "logit"))
ord_curve(model1, thr = x2) # set the threshold as a covariate

## Poisson example of ordered curve
n <- 500
set.seed(1234)
x1 <- rnorm(n)
x2 <- rnorm(n)
beta0 <- 0
beta1 <- 2
beta2 <- 1
lambda1 <- exp(beta0 + beta1 * x1 + beta2 * x2)

y <- rpois(n, lambda1)

## True Model
poismodel1 <- glm(y ~ x1 + x2, family = poisson(link = "log"))
ord_curve(poismodel1, thr = poismodel1$fitted.values)

## Missing a covariate
poismodel2 <- glm(y ~ x1, family = poisson(link = "log"))
ord_curve(poismodel2, thr = poismodel2$fitted.values)
ord_curve(poismodel2, thr = x2)
```

---

qqresid *QQ-plots of DPIT residuals*

Description

Makes a QQ-plot of the DPIT residuals calculated from `resid_disc()`, `resid_semiconti()` or `resid_zeroinfl()`. The plot should be close to the diagonal if the model is correctly specified. Note that this function does not return residuals. To get both residuals and QQ-plot, use `resid_disc()`, `resid_semiconti()` and `resid_zeroinfl()`.

Usage

```
qqresid(model, scale="normal")
```

Arguments

`model`	Fitted model object (e.g., `glm()`, `glm.nb()`, `zeroinfl()`, and `polr()`)
`scale`	You can choose the scale of the residuals between `normal` and `uniform` scales. The sample quantiles of the residuals are plotted against the theoretical quantiles of a standard normal distribution under the normal scale, and against the theoretical quantiles of a uniform (0,1) distribution under the uniform scale. The defalut scale is `normal`.

Value

A QQ plot.

- x-axis: Theoretical quantiles
- y-axis: Sample quantiles generated by DPIT residuals

See Also

resid_disc(), resid_semiconti(), resid_zeroinfl()

Examples

```
n <- 100
b <- c(2, 1, -2)
x1 <- rnorm(n)
x2 <- rbinom(n, 1, 0.7)
y <- rpois(n, exp(b[1] + b[2] * x1 + b[3] * x2))

m1 <- glm(y ~ x1 + x2, family = poisson)
qqresid(m1, scale = "normal")
qqresid(m1, scale = "uniform")
```

---

resid_2pm *Residuals for regression models with two-part outcomes*

Description

Calculates DPIT proposed residuals for model for semi-continuous outcomes. `resid_2pm` can be used either with `model0` and `model1` or with `part0` and `part1` as arguments.

Usage

```
resid_2pm(model0, model1, y, part0, part1, plot=TRUE, scale = "normal")
```

Arguments

`model0`	Model object for 0 outcomes (e.g., logistic regression)
`model1`	Model object for the continuous part (gamma regression)
`y`	Semicontinuous outcome variables
`part0`	Alternative argument to `model0`. One can supply the sequence of probabilities $P(Y_i = 0),\ i = 1, \ldots, n$.
`part1`	Alternative argument to `model1`. One can fit a regression model on the positive data and supply their probability integral transform. Note that the length of `part1` is the number of positive values in `y` and can be shorter than `part0`.
`plot`	A logical value indicating whether or not to return QQ-plot
`scale`	You can choose the scale of the residuals among `normal` and `uniform` scales. The default scale is `normal`.

Details

The DPIT residuals for regression models with semi-continuous outcomes are

$$\hat{r}_i = \frac{\hat{F}(Y_i|\mathbf{X}_i)}{n} \sum_{j=1}^{n} 1\left(\hat{p}_0(\mathbf{X}_j) \le \hat{F}(Y_i|\mathbf{X}_i)\right), i = 1, \ldots, n,$$

where $\hat{p}_0(\mathbf{X}_i)$ is the fitted probability of zero, and $\hat{F}(\cdot|\mathbf{X}_i)$ is the fitted cumulative distribution function for the ith observation. Furthermore,

$$\hat{F}(y|\mathbf{x}) = \hat{p}_0(\mathbf{x}) + (1 - \hat{p}_0(\mathbf{x}))\, \hat{G}(y|\mathbf{x})$$

where $\hat{G}$ is the fitted cumulative distribution for the positive data.

In two-part models, the probability of zero can be modeled using a logistic regression, `model0`, while the positive observations can be modeled using a gamma regression, `model1`. Users can choose to use different models and supply the resulting probability transforms. `part0` should be the sequence of fitted probabilities of zeros $\hat{p}_0(\mathbf{X}_i),\ i = 1, \ldots, n$. `part1` should be the probability integral transform of the positive part $\hat{G}(Y_i|\mathbf{X}_i)$. Note that the length of `part1` is the number of positive values in `y` and can be shorter than `part0`.

Value

Residuals. If plot=TRUE, also produces a QQ plot.

See Also

`resid_semiconti()`

Examples

```
library(MASS)
n <- 500
beta10 <- 1
beta11 <- -2
beta12 <- -1
beta13 <- -1
beta14 <- -1
beta15 <- -2
x11 <- rnorm(n)
x12 <- rbinom(n, size = 1, prob = 0.4)

p1 <- 1 / (1 + exp(-(beta10 + x11 * beta11 + x12 * beta12)))
lambda1 <- exp(beta13 + beta14 * x11 + beta15 * x12)
y2 <- rgamma(n, scale = lambda1 / 2, shape = 2)
y <- rep(0, n)
u <- runif(n, 0, 1)
ind1 <- which(u >= p1)
y[ind1] <- y2[ind1]

# models as input
mgamma <- glm(y[ind1] ~ x11[ind1] + x12[ind1], family = Gamma(link = "log"))
m10 <- glm(y == 0 ~ x12 + x11, family = binomial(link = "logit"))
resid.model <- resid_2pm(model0 = m10, model1 = mgamma, y = y)

# PIT as input
cdfgamma <- pgamma(y[ind1],
```

```
  scale = mgamma$fitted.values * gamma.dispersion(mgamma),
  shape = 1 / gamma.dispersion(mgamma)
)
p1f <- m10$fitted.values
resid.pit <- resid_2pm(y = y, part0 = p1f, part1 = cdfgamma)
```

`resid_disc` *Residuals for regression models with discrete outcomes*

Description

Calculates the DPIT residuals for regression models with discrete outcomes. Specifically, the model assumption of GLMs with binary, ordinal, Poisson, and negative binomial outcomes can be assessed using `resid_disc()`.

Usage

```
resid_disc(model, plot=TRUE, scale="normal")
```

Arguments

`model`	Model object (e.g., `glm`, `glm.nb`, `polr`)
`plot`	A logical value indicating whether or not to return QQ-plot
`scale`	You can choose the scale of the residuals among `normal` and `uniform` scales. The sample quantiles of the residuals are plotted against the theoretical quantiles of a standard normal distribution under the normal scale, and against the theoretical quantiles of a uniform (0,1) distribution under the uniform scale. The defalut scale is `normal`.

Details

The DPIT residual for the ith observation is defined as follows:

$$\hat{r}(Y_i|X_i) = \hat{G}\bigg(\hat{F}(Y_i|\mathbf{X}_i)\bigg)$$

where

$$\hat{G}(s) = \frac{1}{n-1}\sum_{j=1,j\neq i}^{n}\hat{F}\bigg(\hat{F}^{(-1)}(\mathbf{X}_j)\bigg|\mathbf{X}_j\bigg)$$

and $\hat{F}$ refers to the fitted cumulative distribution function. When `scale="uniform"`, DPIT residuals should closely follow a uniform distribution, otherwise it implies model deficiency. When `scale="normal"`, it applies the normal quantile transformation to the DPIT residuals

$$\Phi^{-1}\left[\hat{r}(Y_i|\mathbf{X}_i)\right], i = 1, \ldots, n.$$

The null pattern is the standard normal distribution in this case.

Check reference for more details.

Value

DPIT residuals. If `plot=TRUE`, also produces a QQ plot.

References

Yang, Lu. "Double Probability Integral Transform Residuals for Regression Models with Discrete Outcomes." arXiv preprint arXiv:2308.15596 (2023).

Examples

```
library(MASS)
n <- 500
set.seed(1234)
## Negative Binomial example
# Covariates
x1 <- rnorm(n)
x2 <- rbinom(n, 1, 0.7)
### Parameters
beta0 <- -2
beta1 <- 2
beta2 <- 1
size1 <- 2
lambda1 <- exp(beta0 + beta1 * x1 + beta2 * x2)
# generate outcomes
y <- rnbinom(n, mu = lambda1, size = size1)

# True model
model1 <- glm.nb(y ~ x1 + x2)
resid.nb1 <- resid_disc(model1, plot = TRUE, scale = "uniform")

# Overdispersion
model2 <- glm(y ~ x1 + x2, family = poisson(link = "log"))
resid.nb2 <- resid_disc(model2, plot = TRUE, scale = "normal")

## Binary example
n <- 500
set.seed(1234)
# Covariates
x1 <- rnorm(n, 1, 1)
x2 <- rbinom(n, 1, 0.7)
# Coefficients
beta0 <- -5
beta1 <- 2
beta2 <- 1
beta3 <- 3
q1 <- 1 / (1 + exp(beta0 + beta1 * x1 + beta2 * x2 + beta3 * x1 * x2))
y1 <- rbinom(n, size = 1, prob = 1 - q1)

# True model
model01 <- glm(y1 ~ x1 * x2, family = binomial(link = "logit"))
resid.bin1 <- resid_disc(model01, plot = TRUE)

# Missing covariates
model02 <- glm(y1 ~ x1, family = binomial(link = "logit"))
resid.bin2 <- resid_disc(model02, plot = TRUE)

## Poisson example
n <- 500
set.seed(1234)
# Covariates
```

```
x1 <- rnorm(n)
x2 <- rbinom(n, 1, 0.7)
# Coefficients
beta0 <- -2
beta1 <- 2
beta2 <- 1
lambda1 <- exp(beta0 + beta1 * x1 + beta2 * x2)
y <- rpois(n, lambda1)

# True model
poismodel1 <- glm(y ~ x1 + x2, family = poisson(link = "log"))
resid.poi1 <- resid_disc(poismodel1, plot = TRUE)

# Enlarge three outcomes
y <- rpois(n, lambda1) + c(rep(0, (n - 3)), c(10, 15, 20))
poismodel2 <- glm(y ~ x1 + x2, family = poisson(link = "log"))
resid.poi2 <- resid_disc(poismodel2, plot = TRUE)

## Ordinal example
n <- 500
set.seed(1234)
# Covariates
x1 <- rnorm(n, mean = 2)
# Coefficient
beta1 <- 3

# True model
p0 <- plogis(1, location = beta1 * x1)
p1 <- plogis(4, location = beta1 * x1) - p0
p2 <- 1 - p0 - p1
genemult <- function(p) {
  rmultinom(1, size = 1, prob = c(p[1], p[2], p[3]))
}
test <- apply(cbind(p0, p1, p2), 1, genemult)
y1 <- rep(0, n)
y1[which(test[1, ] == 1)] <- 0
y1[which(test[2, ] == 1)] <- 1
y1[which(test[3, ] == 1)] <- 2
multimodel <- polr(as.factor(y1) ~ x1, method = "logistic")
resid.ord1 <- resid_disc(multimodel, plot = TRUE)

## Non-Proportionality
n <- 500
set.seed(1234)
x1 <- rnorm(n, mean = 2)
beta1 <- 3
beta2 <- 1
p0 <- plogis(1, location = beta1 * x1)
p1 <- plogis(4, location = beta2 * x1) - p0
p2 <- 1 - p0 - p1
genemult <- function(p) {
  rmultinom(1, size = 1, prob = c(p[1], p[2], p[3]))
}
test <- apply(cbind(p0, p1, p2), 1, genemult)
y1 <- rep(0, n)
y1[which(test[1, ] == 1)] <- 0
y1[which(test[2, ] == 1)] <- 1
```

```
y1[which(test[3, ] == 1)] <- 2
multimodel <- polr(as.factor(y1) ~ x1, method = "logistic")
resid.ord2 <- resid_disc(multimodel, plot = TRUE)
```

`resid_quasi` *Quasi Emprical residuals functions*

Description

Draw the QQ-plot for regression models with discrete outcomes using the quasi-empirical residual distribution functions. Specifically, the model assumption of GLMs with binary, ordinal, Poisson, negative binomial, zero-inlated Poisson, and zero-inflated negative binomial outcomes can be applicable to `resid_quasi()`.

Usage

```
resid_quasi(model)
```

Arguments

`model`	Model object (e.g., `glm`, `glm.nb`, `polr`, `zeroinfl`)

Details

The quasi-empirical residual distribution function is defined as follows:

$$\hat{U}(s;\beta) = \sum_{i=1}^{n} W_n(s;\mathbf{X}_i,\beta) 1[F(Y_i|X_i) < H(s;X_i)]$$

where

$$W_n(s;\mathbf{X}_i,\beta) = \frac{K[(H(s;\mathbf{X}_i) - s)/\epsilon_n]}{\sum_{j=1}^{n} K[(H(s;\mathbf{X}_j) - s)/\epsilon_n]}$$

and K is a bounded, symmetric, and Lipschitz continuous kernel.

Value

A QQ plot.

- x-axis: Theoretical quantiles
- y-axis: Sample quantiles

References

Lu Yang (2021). Assessment of Regression Models with Discrete Outcomes Using Quasi-Empirical Residual Distribution Functions, Journal of Computational and Graphical Statistics, 30(4), 1019-1035.

Examples

```
## Negative Binomial example
library(MASS)
# Covariates
n <- 500
x1 <- rnorm(n)
x2 <- rbinom(n, 1, 0.7)
### Parameters
beta0 <- -2
beta1 <- 2
beta2 <- 1
size1 <- 2
lambda1 <- exp(beta0 + beta1 * x1 + beta2 * x2)
# generate outcomes
y <- rnbinom(n, mu = lambda1, size = size1)

# True model
model1 <- glm.nb(y ~ x1 + x2)
resid.nb1 <- resid_quasi(model1)

# Overdispersion
model2 <- glm(y ~ x1 + x2, family = poisson(link = "log"))
resid.nb2 <- resid_quasi(model2)

## Zero inflated Poisson example
library(pscl)
n <- 500
set.seed(1234)
# Covariates
x1 <- rnorm(n)
x2 <- rbinom(n, 1, 0.7)
# Coefficients
beta0 <- -2
beta1 <- 2
beta2 <- 1
beta00 <- -2
beta10 <- 2

# Mean of Poisson part
lambda1 <- exp(beta0 + beta1 * x1 + beta2 * x2)
# Excess zero probability
p0 <- 1 / (1 + exp(-(beta00 + beta10 * x1)))
## simulate outcomes
y0 <- rbinom(n, size = 1, prob = 1 - p0)
y1 <- rpois(n, lambda1)
y <- ifelse(y0 == 0, 0, y1)
## True model
modelzero1 <- zeroinfl(y ~ x1 + x2 | x1, dist = "poisson", link = "logit")
resid.zero1 <- resid_quasi(modelzero1)
```

`resid_semiconti` *Residuals for regression models with semicontinuous outcomes*

Description

Calculates the DPIT residuals for regression models with semi-continuous outcomes. The semi-continuous regression model such as a Tweedie regression model from `tweedie` package or a Tobit regression model from `VGAM`, `AER` packages is used in this function.

Usage

```
resid_semiconti(model, plot=TRUE, scale = "normal")
```

Arguments

`model`	Model object (e.g., `tweedie`, `vglm`, and `tobit`)
`plot`	A logical value indicating whether or not to return QQ-plot
`scale`	You can choose the scale of the residuals between `normal` and `uniform` scales. The default scale is `normal`.

Details

The DPIT residual for the ith semicontinuous observation is defined as follows:

$$\hat{r}_i = \frac{\hat{F}(Y_i|X_i)}{n} \sum_{j=1}^{n} I\left(\hat{p}_0(X_j) \leq \hat{F}(Y_i|X_i)\right),$$

which has a null distribution of uniformity. $\hat{F}$ refers to the fitted cumulative distribution function, and $\hat{p}_0$ refers to the fitted probability of being zero.

Value

Residuals. If plot=TRUE, also produces a QQ plot.

References

Lu Yang (2024). Diagnostics for Regression Models with Semicontinuous Outcomes, Biometrics, https://arxiv.org/abs/2401.06347

See Also

`resid_2pm()`

Examples

```
## Tweedie model
library(tweedie)
library(statmod)
n <- 500
x11 <- rnorm(n)
x12 <- rnorm(n)
beta0 <- 5
beta1 <- 1
beta2 <- 1
lambda1 <- exp(beta0 + beta1 * x11 + beta2 * x12)
y1 <- rtweedie(n, mu = lambda1, xi = 1.6, phi = 10)
# Choose parameter p
# True model
```

```
model1 <-
  glm(y1 ~ x11 + x12,
    family = tweedie(var.power = 1.6, link.power = 0)
  )
resid.tweedie <- resid_semiconti(model1)

## Tobit regression model
library(VGAM)
beta13 <- 1
beta14 <- -3
beta15 <- 3

set.seed(1234)
x11 <- runif(n)
x12 <- runif(n)
lambda1 <- beta13 + beta14 * x11 + beta15 * x12
sd0 <- 0.3
yun <- rnorm(n, mean = lambda1, sd = sd0)
y <- ifelse(yun >= 0, yun, 0)

# Using VGAM package
# True model
fit1 <- vglm(formula = y ~ x11 + x12, tobit(Upper = Inf, Lower = 0, lmu = "identitylink"))
# Missing covariate
fit1miss <- vglm(formula = y ~ x11, tobit(Upper = Inf, Lower = 0, lmu = "identitylink"))

resid.tobit1 <- resid_semiconti(fit1, plot = TRUE)
resid.tobit2 <- resid_semiconti(fit1miss, plot = TRUE)

# Using AER package
library(AER)
# True model
fit2 <- tobit(y ~ x11 + x12, left = 0, right = Inf, dist = "gaussian")
# Missing covariate
fit2miss <- tobit(y ~ x11, left = 0, right = Inf, dist = "gaussian")
resid.aer1 <- resid_semiconti(fit2, plot = TRUE)
resid.aer2 <- resid_semiconti(fit2miss, plot = TRUE)
```

`resid_zeroinfl` *Residuals for regression models with zero-inflated outcomes*

Description

Caluates the DPIT residuals for a regression model with zero-inflated discrete outcome. A zero-inflated model from pscl is used in this function.

Usage

```
resid_zeroinfl(model, plot=TRUE, scale='normal')
```

Arguments

model	Model object, which is the output of pscl::zeroinfl.
plot	A logical value indicating whether or not to return QQ-plot.

scale — You can choose the scale of the residuals among normal and uniform scales. The default scale is normal.

Value

DPIT residuals. If plot=TRUE, also produces a QQ plot.

References

Yang, Lu. "Double Probability Integral Transform Residuals for Regression Models with Discrete Outcomes." arXiv preprint arXiv:2308.15596 (2023).

Examples

```
## Zero-Inflated Poisson
library(pscl)
n <- 500
set.seed(1234)
# Covariates
x1 <- rnorm(n)
x2 <- rbinom(n, 1, 0.7)
# Coefficients
beta0 <- -2
beta1 <- 2
beta2 <- 1
beta00 <- -2
beta10 <- 2

# Mean of Poisson part
lambda1 <- exp(beta0 + beta1 * x1 + beta2 * x2)
# Excess zero probability
p0 <- 1 / (1 + exp(-(beta00 + beta10 * x1)))
## simulate outcomes
y0 <- rbinom(n, size = 1, prob = 1 - p0)
y1 <- rpois(n, lambda1)
y <- ifelse(y0 == 0, 0, y1)
## True model
modelzero1 <- zeroinfl(y ~ x1 + x2 | x1, dist = "poisson", link = "logit")
resid.zero1 <- resid_zeroinfl(modelzero1, plot = TRUE, scale = "uniform")

## Zero inflation
modelzero2 <- glm(y ~ x1 + x2, family = poisson(link = "log"))
resid.zero2 <- resid_disc(modelzero2, plot = TRUE, scale = "normal")
```

Index