Package 'assessor'

September 13, 2023

Title Assessment tools for regression models with discrete and semicontinuous outcomes

Version 1.0.0

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 transorm (DPIT) residuals, constructs QQ plots of residuals, and the ordered curve for assessing mean structures.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

URL https://github.com/jhlee1408/residuals

BugReports https://github.com/jhlee1408/residuals/issues

Imports tweedie

Suggests glmnet, knitr, MASS, pscl, rmarkdown, statmod

VignetteBuilder rmarkdown

R topics documented:

`ord_curve` *Ordered Curve*

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., glm, glm.nb, polr)
thr	threshold variable (e.g., predictor, fitted values, or variable to be included as a covariate)

Details

The ordered curve plots

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

against

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

, 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, $(\hat{L}_1(t), \hat{L}_2(t))$ 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(eg. x1), a linear combination of predictor(eg.x1+x2), or fitted values(eg. 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.

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)

model0 <- glm(y1~x1*x2,family =binomial(link = "logit") )
ord_curve(model0,thr=model0$fitted.values) # set the threshold as fitted values
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)
poismodel1 <- glm(y ~ x1+x2, family = poisson(link = "log"))
ord_curve(poismodel1,thr=poismodel1$fitted.values)

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

`qqresid` *qqplot with 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 qqplot 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`.

See Also

`resid_disc()`, `resid_semiconti()`, `resid_zeroinfl()`

Examples

```
n <- 1e2
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") ## qqplot of poisson regression
qqresid(m1, scale="uniform")
```

`resid_disc` *Residuals for discrete outcome regression*

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 qqplot 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:

$$\hat{r}(Y_i|X_i) = \hat{G}_{M_i}\left(\hat{F}_M(Y_i|\mathbf{X}_i)\right)$$

$$\text{where } \hat{G}_{M_i}(s) = \frac{1}{n-1}\sum_{j=1,j\neq i}^{n} \hat{F}_M\left(\hat{F}_M^{(-1)}(\mathbf{X}_j)\middle|\mathbf{X}_j\right)$$

where $\hat{F}_M$ 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_disc(model1,plot = TRUE, scale="uniform")

# Overdispersion
model2 <- glm(y~x1+x2,family = poisson(link = "log"))
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_disc(model01,plot = TRUE)

# Missing covariates
model02 <- glm(y1~x1,family =binomial(link = "logit") )
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_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_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_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_disc(multimodel,plot = TRUE)
```

`resid_semiconti` *Residuals for semicontinuous outcome regression*

Description

`resid.semiconti` is used to calculate newly proposed residuals for semi-continuous outcomes regression such as tweedie model. A model object of semicontinuous regression from `tweedie` package is recommended.

Usage

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

Arguments

model	model object(using tweedie family)
plot	A logical value indicating whether or not to return QQ-plot
scale	You can choose the scale of residuals among normal and uniform scales. The defalut scale is normal.

Details

The proposed residuals are defined as

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

, which has a null distribution of uniformity.

Value

The double probability integral transform residuals(DPIT residuals).

---

resid_zeroinfl *Residuals for zero-inflated regression model*

Description

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

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 qqplot 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_zeroinfl(modelzero1,plot=TRUE, scale="uniform")

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