Package ‘glmMisrep’

April 18, 2024

Type Package

Title Generalized Linear Models Adjusting for Misrepresentation

Version 0.1.1

Depends R (>= 3.5.0)

Description Fit Generalized Linear Models to continuous and count outcomes, as well as estimate the prevalence of misrepresentation of an important binary predictor. Misrepresentation typically arises when there is an incentive for the binary factor to be misclassified in one direction (e.g., in insurance settings where policy holders may purposely deny a risk status in order to lower the insurance premium). This is accomplished by treating a subset of the response variable as resulting from a mixture distribution. Model parameters are estimated via the Expectation Maximization algorithm and standard errors of the estimates are obtained from closed forms of the Observed Fisher Information. For an introduction to the models and the misrepresentation framework, see Xia et. al., (2023) <https://variancejournal.org/article/73151-maximum-likelihood-approaches-to-misrepresentation-models-in-glm-ratemaking-model-compariso

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports MASS, poisson.glm.mix, stats

NeedsCompilation no

Author Patrick Rafael [cre, aut],
Xia Michelle [aut],
Rexford Akakpo [aut]

Maintainer Patrick Rafael <pbr2608@vt.edu>

Repository CRAN

Date/Publication 2024-04-18 17:43:06 UTC

R topics documented:

`gammaRegMisrepEM` *Fit a Gamma Misrepresentation Model using EM Algorithm*

Description

`gammaRegMisrepEM` is used to fit a Gamma regression model, adjusting for misrepresentation on a binary predictor. The function uses the Expectation Maximization algorithm and allows multiple additional correctly measured independent variables in the Gamma regression with a log-link function that is typically used in insurance claims modeling. Standard errors of model estimates are obtained from closed form expressions of the Observed Fisher Information.

Usage

```
gammaRegMisrepEM(formula, v_star, data, lambda = c(0.6,0.4),
                 epsilon = 1e-08, maxit = 10000,
                 maxrestarts = 20, verb = FALSE)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
`v_star`	a character specifying the name of the binary predictor that is suspected of being misrepresented.
`data`	a dataframe containing the variables in the model.
`lambda`	initial mixing proportions used to start the EM algorithm. A numeric vector of length two, with the second element being the prevalence of misrepresentation.
`epsilon`	tolerance for convergence. Convergence is reached when the log-likelihood increases by less than epsilon.
`maxit`	the maximum number of iterations the EM routine will run for.
`maxrestarts`	how many times the EM routine will attempt to converge. When conergence is not achieved, the EM routine restarts with new randomly selected mixing proportions.
`verb`	logical. If TRUE, the difference in new .vs. old log-likelihood and the current log-likelihood is printed to the console after every iteration. If TRUE, the user will also be notifed if the EM algorithm must restart with new mixing proportions.

Details

Models for gammaRegMisrepEM are specified symbolically. Like the lm and glm functions, the model has the form response ~ terms, where response is the numeric response vector and terms is a series of terms which specifies a linear predictor for response.

Currently, formula specification can accommodate the following expressions:

- transformations of the response: log(y) ~ x
- polynomial terms: y ~ x + I(x^2)
- interactions: y ~ x*z

Including an offset term (e.g. y ~ x + offset()) is currently not supported.

Value

gammaRegMisrepEM returns an object of class "misrepEM".

The function summary is used to obtain and print a summary of the results.

An object of class "misrepEM" is a list containing the following 14 elements:

y	the response used.
lambda	numeric. The estimated prevalence of misrepresentation.
params	a numeric vector containing the estimated parameters.
loglik	the final maximized log-likelihood.
posterior	a numeric vector. The posterior probability that the *i-th* observation is not misrepresented for observations where the suspected misrepresented variable is zero, based on the last iteration of the EM algorithm. The values are not meaningful for observations where the suspected misrepresented variable is one.
all.loglik	a numeric vector containing the log-likelihood at every iteration.
cov.estimates	the inverse of the observed fisher information matrix evaluated at the maximum likelihood estimates.
std.error	a numeric vector containing the standard errors of regression coefficients.
t.values	a numeric vector containing the standardized regression coefficients.
p.values	a numeric vector containing the *p*-values of the regression coefficients.
ICs	a numeric vector of length three containing the AIC, AICc, and BIC.
ft	a character containing the name of the function.
formula	an object of class formula indicating the model that was fit.
v_star_name	a character containing the name of the binary predictor suspected of misrepresentation.

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Akakpo, R. M., Xia, M., & Polansky, A. M. (2019). Frequentist inference in insurance ratemaking models adjusting for misrepresentation. *ASTIN Bulletin: The Journal of the IAA, 49*(1), 117-146.

Xia, M., Hua, L., & Vadnais, G. (2018). Embedded predictive analysis of misrepresentation risk in GLM ratemaking models. *Variance, 12*(1), 39-58.

Examples

```
set.seed(314159)

# Simulate data
n <- 1000
p0 <- 0.25

X1 <- rbinom(n, 1, 0.4)
X2 <- sample(x = c("a", "b", "c"), size = n, replace = TRUE)
X3 <- rnorm(n, 0, 1)

theta0 <- 0.3
V <- rbinom(n,1,theta0)
V_star <- V
V_star[V==1] <- rbinom(sum(V==1),1,1-p0)

a0 <- 1
a1 <- 2
a2 <- 0
a3 <- -1
a4 <- 4
a5 <- 2

mu <- rep(0, n)

for(i in 1:n){

  mu[i] <- exp(a0 + a1*X1 + a4*X3 + a5*V )[i]

  if(X2[i] == "a" || X2[i] == "b"){

    mu[i] <- mu[i]*exp(a2)

  }else{
    mu[i] <- mu[i]*exp(a3)
  }

}

phi <- 0.2
alpha0 <- 1/phi
beta <- 1/mu/phi
Y <- rgamma(n, alpha0, beta)

data <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, V_star = V_star)

# "a" is the reference
data$X2 <- as.factor(data$X2)
```

```
# Model with main effects:
gamma_mod <- gammaRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star,
                              v_star = "V_star", data = data)

# The prevalence of misrepresentation;
(theta0 * p0) / (1 - theta0*(1-p0)) # 0.09677419

# Parameter estimates and estimated prevalence of
# misrepresentation (lambda);
summary(gamma_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.99356    0.03013  32.97245   <2e-16 ***
# X1           2.02152    0.03078  65.68276   <2e-16 ***
# X2b         -0.00679    0.03708  -0.18309  0.85477
# X2c         -1.02578    0.03684 -27.84599   <2e-16 ***
# X3           3.97883    0.01495 266.21973   <2e-16 ***
# V_star       2.00437    0.03107  64.51234   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 5650.696 5650.841 5689.958
# ---
# Log-Likelihood
#      -2817.348
# ---
# Lambda:  0.1083894 std.err:  0.01160662


# Fitting an interaction between X2 and X3;

a6 <- -2
a7 <- 2

for(i in 1:n){

  if(X2[i] == "c"){
    mu[i] <- mu[i]*exp(a6*X3[i])
  }else{
    if(X2[i] =="b"){
      mu[i] <- mu[i]*exp(a7*X3[i])
    }
  }
}

beta <- 1/mu/phi
Y <- rgamma(n, alpha0, beta)

data$Y <- Y

gamma_mod <- gammaRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3,
```

```
                              v_star = "V_star", data = data)

summary(gamma_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.96205    0.03086  31.17145   <2e-16 ***
# X1           2.00411    0.03061  65.46734   <2e-16 ***
# X2b         -0.00987    0.03682  -0.26818  0.78862
# X2c         -0.99957    0.03733 -26.77449   <2e-16 ***
# X3           3.98282    0.02484 160.31083   <2e-16 ***
# V_star       2.01107    0.03077  65.36550   <2e-16 ***
# X2b:X3       1.95884    0.03573  54.82466   <2e-16 ***
# X2c:X3      -1.98595    0.03567 -55.67827   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 5633.984 5634.207 5683.062
# ---
# Log-Likelihood
#      -2806.992
# ---
# Lambda:  0.1131951 std.err:  0.01181678


# Model fitting with a polynomial effect;

a8 <- -0.5

mu <- mu*exp(a8*X3^2)

beta <- 1/mu/phi
Y <- rgamma(n, alpha0, beta)

data$Y <- Y

gamma_mod <- gammaRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3 + I(X3^2),
                              v_star = "V_star", data = data)

summary(gamma_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  1.04312    0.03164  32.96624   <2e-16 ***
# X1           2.04411    0.02929  69.79020   <2e-16 ***
# X2b         -0.10418    0.03512  -2.96620  0.00309  **
# X2c         -1.08910    0.03531 -30.84683   <2e-16 ***
# X3           4.00265    0.02421 165.31001   <2e-16 ***
# V_star       1.98741    0.02951  67.35719   <2e-16 ***
# I(X3^2)     -0.51152    0.01350 -37.90112   <2e-16 ***
# X2b:X3       1.98709    0.03598  55.22750   <2e-16 ***
# X2c:X3      -2.03395    0.03692 -55.09491   <2e-16 ***
```

```
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 4559.969 4560.236 4613.954
# ---
# Log-Likelihood
#      -2268.984
# ---
# Lambda:  0.111464 std.err:  0.01173143
```

`LnRegMisrepEM` *Fit a Lognormal Misrepresentation Model using EM Algorithm*

Description

`LnRegMisrepEM` is used to fit a Lognormal regression model, adjusting for misrepresentation on a binary predictor. The function uses the Expectation Maximization algorithm and allows multiple additional correctly measured independent variables in the Lognormal regression with an identity link function that is typically used in insurance claims modeling. Standard errors of model estimates are obtained from closed form expressions of the Observed Fisher Information.

Usage

```
LnRegMisrepEM(formula, v_star, data, lambda = c(0.6,0.4),
                  epsilon = 1e-08, maxit = 10000,
                  maxrestarts = 20, verb = FALSE)
```

Arguments

`formula`	an object of class "`formula`" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
`v_star`	a character specifying the name of the binary predictor that is suspected of being misrepresented.
`data`	a dataframe containing the variables in the model.
`lambda`	initial mixing proportions used to start the EM algorithm. A numeric vector of length two, with the second element being the prevalence of misrepresentation.
`epsilon`	tolerance for convergence. Convergence is reached when the log-likelihood increases by less than epsilon.
`maxit`	the maximum number of iterations the EM routine will run for.
`maxrestarts`	how many times the EM routine will attempt to converge. When conergence is not achieved, the EM routine restarts with new randomly selected mixing proportions.

verb	logical. If TRUE, the difference in new .vs. old log-likelihood and the current log-likelihood is printed to the console after every iteration. If TRUE, the user will also be notifed if the EM algorithm must restart with new mixing proportions.

Details

Please note: In the Log-Normal regression setting, the response is assumed to be Log-Normally distributed, so the function `LnRegMisrepEM` requires that the `formula` argument have a certain form: `log(response) ~ terms`. See 'Examples' for a demonstration.

Models for `LnRegMisrepEM` are specified symbolically. Like the `lm` and `glm` functions, the model has the form `response ~ terms`, where `response` is the numeric response vector and `terms` is a series of terms which specifies a linear predictor for `response`.

Currently, formula specification can accommodate the following expressions:

- transformations of the response: `log(y) ~ x`
- polynomial terms: `y ~ x + I(x^2)`
- interactions: `y ~ x*z`

Including an offset term (e.g. `y ~ x + offset()`) is currently not supported.

Value

`LnRegMisrepEM` returns an object of class "misrepEM".

The function `summary` is used to obtain and print a summary of the results.

An object of class "misrepEM" is a list containing the following 14 elements:

y	the response used.
lambda	numeric. The estimated prevalence of misrepresentation.
params	a numeric vector containing the estimated parameters.
loglik	the final maximized log-likelihood.
posterior	a numeric vector. The posterior probability that the *i-th* observation is not misrepresented for observations where the suspected misrepresented variable is zero, based on the last iteration of the EM algorithm. The values are not meaningful for observations where the suspected misrepresented variable is one.
all.loglik	a numeric vector containing the log-likelihood at every iteration.
cov.estimates	the inverse of the observed fisher information matrix evaluated at the maximum likelihood estimates.
std.error	a numeric vector containing the standard errors of regression coefficients.
t.values	a numeric vector containing the standardized regression coefficients.
p.values	a numeric vector containing the p-values of the regression coefficients.
ICs	a numeric vector of length three containing the AIC, AICc, and BIC.
ft	a character containing the name of the function.
formula	an object of class `formula` indicating the model that was fit.
v_star_name	a character containing the name of the binary predictor suspected of misrepresentation.

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Akakpo, R. M., Xia, M., & Polansky, A. M. (2019). Frequentist inference in insurance ratemaking models adjusting for misrepresentation. *ASTIN Bulletin: The Journal of the IAA, 49*(1), 117-146.

Xia, M., Hua, L., & Vadnais, G. (2018). Embedded predictive analysis of misrepresentation risk in GLM ratemaking models. *Variance, 12*(1), 39-58.

Examples

```
# Simulate data
n <- 1000
p0 <- 0.25

X1 <- rbinom(n, 1, 0.4)
X2 <- sample(x = c("a", "b", "c"), size = n, replace = TRUE)
X3 <- rnorm(n, 0, 1)

theta0 <- 0.3
V <- rbinom(n,1,theta0)
V_star <- V
V_star[V==1] <- rbinom(sum(V==1),1,1-p0)

a0 <- 1
a1 <- 2
a2 <- 0
a3 <- -1
a4 <- 4
a5 <- 2

mu <- rep(0, n)

for(i in 1:n){

  mu[i] <- exp(a0 + a1*X1 + a4*X3 + a5*V )[i]

  if(X2[i] == "a" || X2[i] == "b"){

    mu[i] <- mu[i]*exp(a2)

  }else{
    mu[i] <- mu[i]*exp(a3)
  }

}

sigma <- 0.427
mu.norm <- log(mu)-sigma^2/2
Y <- rlnorm(n, mu.norm, sigma)
```

```
data <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, V_star = V_star)

# "a" is the reference
data$X2 <- as.factor(data$X2)

# Model with main effects:
LN_mod <- LnRegMisrepEM(formula = log(Y) ~ X1 + X2 + X3 + V_star,
                        v_star = "V_star", data = data)

# The prevalence of misrepresentation;
(theta0 * p0) / (1 - theta0*(1-p0)) # 0.09677419

# Parameter estimates and estimated prevalence of
# misrepresentation (lambda);
summary(LN_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  1.00664    0.02874  35.02082   <2e-16 ***
# X1           1.95903    0.02825  69.35263   <2e-16 ***
# X2b          0.04106    0.03413   1.20303  0.22925
# X2c         -1.00367    0.03418 -29.36328   <2e-16 ***
# X3           4.00031    0.01366 292.75312   <2e-16 ***
# V_star       2.01422    0.02922  68.93902   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 5555.224 5555.370 5594.486
# ---
# Log-Likelihood
#      -2769.612
# ---
# Lambda:  0.11085 std.err:  0.01150365


# Fitting an interaction between X2 and X3;

a6 <- -2
a7 <- 2

for(i in 1:n){

  if(X2[i] == "c"){
    mu[i] <- mu[i]*exp(a6*X3[i])
  }else{
    if(X2[i] =="b"){
      mu[i] <- mu[i]*exp(a7*X3[i])
    }
  }
}

mu.norm <- log(mu)-sigma^2/2
```

```
Y <- rlnorm(n, mu.norm, sigma)

data$Y <- Y

LN_mod <- LnRegMisrepEM(formula = log(Y) ~ X1 + X2 + X3 + V_star + X2*X3,
                        v_star = "V_star", data = data)

summary(LN_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.95064    0.02905  32.71943   <2e-16 ***
# X1           2.04258    0.02876  71.02228   <2e-16 ***
# X2b          0.00204    0.03463   0.05879  0.95314
# X2c         -0.97738    0.03469 -28.17315   <2e-16 ***
# X3           3.97014    0.02341 169.61122   <2e-16 ***
# V_star       2.01894    0.02967  68.04786   <2e-16 ***
# X2b:X3       2.00436    0.03459  57.95433   <2e-16 ***
# X2c:X3      -1.97573    0.03431 -57.59173   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 5505.180 5505.402 5554.257
# ---
# Log-Likelihood
#       -2742.59
# ---
# Lambda:  0.1055629 std.err:  0.01134298

# Model fitting with a polynomial effect;

a8 <- -0.5

mu <- mu*exp(a8*X3^2)

mu.norm <- log(mu)-sigma^2/2
Y <- rlnorm(n, mu.norm, sigma)

data$Y <- Y

LN_mod <- LnRegMisrepEM(formula = log(Y) ~ X1 + X2 + X3 + V_star + X2*X3 + I(X3^2),
                        v_star = "V_star", data = data)

summary(LN_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.95591    0.03084  30.99533   <2e-16 ***
# X1           2.00070    0.02878  69.52672   <2e-16 ***
# X2b          0.09309    0.03480   2.67464   0.0076  **
# X2c         -0.96572    0.03455 -27.95530   <2e-16 ***
# X3           3.96765    0.02378 166.82860   <2e-16 ***
```

```
# V_star       2.00513    0.02967  67.58486   <2e-16 ***
# I(X3^2)     -0.49043    0.00983 -49.90063   <2e-16 ***
# X2b:X3       2.04614    0.03454  59.24411   <2e-16 ***
# X2c:X3      -1.97248    0.03383 -58.30378   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 4537.485 4537.752 4591.470
# ---
# Log-Likelihood
#      -2257.742
# ---
# Lambda:  0.1061872 std.err:  0.01138758
```

MEPS14 *MEPS 2014 Full Year Consolidated Data File*

Description

MEPS14 is a subset of the MEPS 2014 Full Year Consolidated Data File, as described in Xia et. al., (2023).

Usage

```
data("MEPS14")
```

Format

A data frame with 13,301 observations on the following 7 variables:

`TOTEXP14` total medical expenditure.

`OBTOTV14` total number of office-based visits.

`UNINS14` uninsured status (1 - insured, 0 - uninsured).

`SEX` sex (1 - male, 0 - female).

`AGE14X` age.

`ADSMOK42` smoking status (1 - yes, 0 - no).

`RTHLTH53` perceieved health status (1 - excellent, 5 - poor).

Source

https://meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-171

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Examples

```
# Reproducing table 4 in Xia et. al., (2023).

data(MEPS14)

colMeans(MEPS14)
#    TOTEXP14     OBTOTV14      UNINS14          SEX       AGE14X     ADSMOK42     RTHLTH53
#5042.4647771    6.2260732    0.1242012    0.4153071   41.6628825    0.1670551    2.4319224

apply(MEPS14, 2, sd)
#    TOTEXP14     OBTOTV14      UNINS14          SEX       AGE14X     ADSMOK42     RTHLTH53
#1.358567e+04 1.272065e+01 3.298233e-01 4.927934e-01 1.332746e+01 3.730391e-01 1.074713e+00

sum(MEPS14$OBTOTV14 == 0 ) / nrow(MEPS14)
# [1] 0.1595369

sd(MEPS14$OBTOTV14 == 0)
# [1] 0.3661898

# Fit Gamma regression model with insured status as
# the misrepresented variable.
MEPS14$RTHLTH53 <- as.factor(MEPS14$RTHLTH53)

gamma_fit <- gammaRegMisrepEM(formula = TOTEXP14 ~ UNINS14
             + SEX + AGE14X + ADSMOK42 + RTHLTH53,
             v_star = "UNINS14", data = MEPS14)

# summary returns a table of summary statistics, including
# goodness of fits (AIC, AICc, BIC), as well as the
# estimated prevalence of misrepresentation.
summary(gamma_fit)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  8.03379    0.05341 150.41937   <2e-16 ***
# UNINS14     -1.98132    0.03170 -62.49292   <2e-16 ***
# SEX         -0.20427    0.02669  -7.65320   <2e-16 ***
# AGE14X       0.02764    0.00099  27.83485   <2e-16 ***
# ADSMOK42    -0.08868    0.03653  -2.42776  0.01521   *
# RTHLTH532    0.24923    0.03533   7.05469   <2e-16 ***
# RTHLTH533    0.53860    0.03655  14.73488   <2e-16 ***
# RTHLTH534    1.00615    0.04837  20.80026   <2e-16 ***
# RTHLTH535    1.87845    0.08104  23.17833   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 241083.9 241083.9 241166.3
# ---
# Log-Likelihood
```

```
#      -120530.9
# ---
# Lambda:  0.7734337 std.err:  0.009628053


# Fit Lognormal regression model with insured status as
# the misrepresented variable.
LN_fit <- LnRegMisrepEM(formula = log(TOTEXP14) ~ UNINS14
             + SEX + AGE14X + ADSMOK42 + RTHLTH53,
             v_star = "UNINS14", data = MEPS14)

summary(LN_fit)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  7.28974    0.05648 129.05986   <2e-16 ***
# UNINS14     -1.29503    0.05496 -23.56317   <2e-16 ***
# SEX         -0.29590    0.02808 -10.53844   <2e-16 ***
# AGE14X       0.02460    0.00107  23.10180   <2e-16 ***
# ADSMOK42    -0.07008    0.03756  -1.86591  0.06208   .
# RTHLTH532    0.26349    0.03831   6.87786   <2e-16 ***
# RTHLTH533    0.47184    0.03942  11.97017   <2e-16 ***
# RTHLTH534    1.05065    0.04990  21.05580   <2e-16 ***
# RTHLTH535    1.94978    0.08067  24.16987   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 239726.4 239726.4 239808.8
# ---
# Log-Likelihood
#      -119852.2
# ---
# Lambda:  0.1110631 std.err:  0.02548188


# Fit Negative Binomial regression model with insured status as
# the misrepresented variable.
NB_fit <- nbRegMisrepEM(formula = OBTOTV14 ~ UNINS14
             + SEX + AGE14X + ADSMOK42 + RTHLTH53,
             v_star = "UNINS14", data = MEPS14)


summary(NB_fit)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  2.00472    0.05463  36.69491   <2e-16 ***
# UNINS14     -1.68638    0.03371 -50.02640   <2e-16 ***
# SEX         -0.40917    0.02303 -17.76536   <2e-16 ***
# AGE14X       0.01897    0.00087  21.91823   <2e-16 ***
# ADSMOK42    -0.11391    0.03038  -3.74948  0.00018 ***
# RTHLTH532    0.20720    0.03183   6.50966   <2e-16 ***
```

```
# RTHLTH533    0.36794    0.03240  11.35678   <2e-16 ***
# RTHLTH534    0.72357    0.03978  18.18859   <2e-16 ***
# RTHLTH535    1.24468    0.06281  19.81714   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 72788.71 72788.73 72871.16
# ---
# Log-Likelihood
#      -36383.35
# ---
# Lambda:  0.8351591 std.err:  0.009627158


# Fit Poisson regression model with smoking status as
# the misrepresented variable.
pois_fit <- poisRegMisrepEM(formula = OBTOTV14 ~ UNINS14
             + SEX + AGE14X + ADSMOK42 + RTHLTH53,
             v_star = "UNINS14", data = MEPS14)

summary(pois_fit)

# Coefficients:
#             Estimate Std. Error    z value Pr(>|z|)
# (Intercept)  2.27367    0.02276   99.87676   <2e-16 ***
# UNINS14     -2.03719    0.00730 -279.00809   <2e-16 ***
# SEX         -0.18594    0.01090  -17.05204   <2e-16 ***
# AGE14X       0.01631    0.00042   38.90467   <2e-16 ***
# ADSMOK42     0.09594    0.01313    7.30930   <2e-16 ***
# RTHLTH532    0.14918    0.01641    9.09033   <2e-16 ***
# RTHLTH533    0.31282    0.01620   19.31078   <2e-16 ***
# RTHLTH534    0.75044    0.01793   41.85270   <2e-16 ***
# RTHLTH535    1.09859    0.02265   48.49410   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 99599.31 99599.33 99674.27
# ---
# Log-Likelihood
#      -49789.66
# ---
# Lambda:  0.85957 std.err:  0.00348128
```

nbRegMisrepEM *Fit a Negative Binomial Misrepresentation Model using EM Algorithm*

Description

nbRegMisrepEM is used to fit a Negative Binomial regression model, adjusting for misrepresentation on a binary predictor. The function uses the Expectation Maximization algorithm and allows multiple additional correctly measured independent variables in the Negative Binomial regression with a log-link function that is typically used in insurance claims modeling. Standard errors of model estimates are obtained from closed form expressions of the Observed Fisher Information.

Usage

```
nbRegMisrepEM(formula, v_star, data, lambda = c(0.6,0.4),
                 epsilon = 1e-08, maxit = 10000,
                 maxrestarts = 20, verb = FALSE)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
v_star	a character specifying the name of the binary predictor that is suspected of being misrepresented.
data	a dataframe containing the variables in the model.
lambda	initial mixing proportions used to start the EM algorithm. A numeric vector of length two, with the second element being the prevalence of misrepresentation.
epsilon	tolerance for convergence. Convergence is reached when the log-likelihood increases by less than epsilon.
maxit	the maximum number of iterations the EM routine will run for.
maxrestarts	how many times the EM routine will attempt to converge. When conergence is not achieved, the EM routine restarts with new randomly selected mixing proportions.
verb	logical. If TRUE, the difference in new .vs. old log-likelihood and the current log-likelihood is printed to the console after every iteration. If TRUE, the user will also be notifed if the EM algorithm must restart with new mixing proportions.

Details

Models for nbRegMisrepEM are specified symbolically. Like the lm and glm functions, the model has the form response ~ terms, where response is the numeric response vector and terms is a series of terms which specifies a linear predictor for response.

Currently, formula specification can accommodate the following expressions:

- transformations of the response: log(y) ~ x
- polynomial terms: y ~ x + I(x^2)
- interactions: y ~ x*z

Including an offset term (e.g. y ~ x + offset()) is currently not supported.

Value

`nbRegMisrepEM` returns an object of class "misrepEM".

The function `summary` is used to obtain and print a summary of the results.

An object of class "misrepEM" is a list containing the following 14 elements:

`y`	the response used.
`lambda`	numeric. The estimated prevalence of misrepresentation.
`params`	a numeric vector containing the estimated parameters.
`loglik`	the final maximized log-likelihood.
`posterior`	a numeric vector. The posterior probability that the *i-th* observation is not misrepresented for observations where the suspected misrepresented variable is zero, based on the last iteration of the EM algorithm. The values are not meaningful for observations where the suspected misrepresented variable is one.
`all.loglik`	a numeric vector containing the log-likelihood at every iteration.
`cov.estimates`	the inverse of the observed fisher information matrix evaluated at the maximum likelihood estimates.
`std.error`	a numeric vector containing the standard errors of regression coefficients.
`t.values`	a numeric vector containing the standardized regression coefficients.
`p.values`	a numeric vector containing the *p*-values of the regression coefficients.
`ICs`	a numeric vector of length three containing the AIC, AICc, and BIC.
`ft`	a character containing the name of the function.
`formula`	an object of class `formula` indicating the model that was fit.
`v_star_name`	a character containing the name of the binary predictor suspected of misrepresentation.

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Akakpo, R. M., Xia, M., & Polansky, A. M. (2019). Frequentist inference in insurance ratemaking models adjusting for misrepresentation. *ASTIN Bulletin: The Journal of the IAA, 49*(1), 117-146.

Xia, M., Hua, L., & Vadnais, G. (2018). Embedded predictive analysis of misrepresentation risk in GLM ratemaking models. *Variance, 12*(1), 39-58.

Examples

```
set.seed(314159)

# Simulate data
n <- 1000
p0 <- 0.25
```

```
X1 <- rbinom(n, 1, 0.4)
X2 <- sample(x = c("a", "b", "c"), size = n, replace = TRUE)
X3 <- rnorm(n, 0, 1)

theta0 <- 0.3
V <- rbinom(n,1,theta0)
V_star <- V
V_star[V==1] <- rbinom(sum(V==1),1,1-p0)

a0 <- 1
a1 <- 2
a2 <- 0
a3 <- -1
a4 <- 4
a5 <- 2

mu <- rep(0, n)

for(i in 1:n){

  mu[i] <- exp(a0 + a1*X1 + a4*X3 + a5*V )[i]

  if(X2[i] == "a" || X2[i] == "b"){

    mu[i] <- mu[i]*exp(a2)

  }else{
    mu[i] <- mu[i]*exp(a3)
  }

}

Y <- rnbinom(n, size = 1, mu = mu)

data <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, V_star = V_star)

# "a" is the reference
data$X2 <- as.factor(data$X2)

# Model with main effects:
NB_mod <- nbRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star,
                        v_star = "V_star", data = data)

# The prevalence of misrepresentation;
(theta0 * p0) / (1 - theta0*(1-p0)) # 0.09677419

# Parameter estimates and estimated prevalence of
# misrepresentation (lambda);
summary(NB_mod)

# Coefficients:
#             Estimate Std. Error  t value Pr(>|t|)
# (Intercept)  0.94091    0.10797  8.71423   <2e-16 ***
```

```
# X1           2.03485    0.09517 21.38182   <2e-16 ***
# X2b          0.13346    0.10998  1.21356  0.22521
# X2c         -0.96514    0.11629 -8.29914   <2e-16 ***
# X3           4.07667    0.05874 69.40599   <2e-16 ***
# V_star       1.90011    0.09517 19.96485   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 7661.457 7661.602 7700.719
# ---
# Log-Likelihood
#      -3822.728
# ---
# Lambda:  0.093119 std.err:  0.02233344


# Fitting an interaction between X2 and X3;

a6 <- -2
a7 <- 2

for(i in 1:n){

  if(X2[i] == "c"){
    mu[i] <- mu[i]*exp(a6*X3[i])
  }else{
    if(X2[i] =="b"){
      mu[i] <- mu[i]*exp(a7*X3[i])
    }
  }
}

Y <- rnbinom(n, size = 1, mu = mu)

data$Y <- Y

NB_mod <- nbRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3,
                        v_star = "V_star", data = data)

summary(NB_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.89452    0.11135   8.03331   <2e-16 ***
# X1           2.13269    0.08473  25.17143   <2e-16 ***
# X2b         -0.01559    0.12545  -0.12429  0.90111
# X2c         -0.95827    0.11665  -8.21469   <2e-16 ***
# X3           4.09454    0.09061  45.19049   <2e-16 ***
# V_star       2.08187    0.08503  24.48402   <2e-16 ***
# X2b:X3       1.84705    0.13130  14.06693   <2e-16 ***
# X2c:X3      -2.11044    0.11910 -17.72024   <2e-16 ***
# ---
```

```
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 7740.111 7740.334 7789.189
# ---
# Log-Likelihood
#      -3860.056
# ---
# Lambda:  0.08479587 std.err:  0.01901557


# Model fitting with a polynomial effect;

a8 <- -0.5

mu <- mu*exp(a8*X3^2)

Y <- rnbinom(n, size = 1, mu = mu)

data$Y <- Y

NB_mod <- nbRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3 + I(X3^2),
                        v_star = "V_star", data = data)

summary(NB_mod)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.96498    0.11201   8.61478   <2e-16 ***
# X1           2.09647    0.09310  22.51926   <2e-16 ***
# X2b         -0.02546    0.13341  -0.19082   0.8487
# X2c         -1.08524    0.12751  -8.51091   <2e-16 ***
# X3           4.03397    0.11939  33.78945   <2e-16 ***
# V_star       1.99765    0.09395  21.26217   <2e-16 ***
# I(X3^2)     -0.49023    0.05312  -9.22849   <2e-16 ***
# X2b:X3       2.00513    0.14127  14.19333   <2e-16 ***
# X2c:X3      -1.93432    0.13657 -14.16309   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 7181.267 7181.535 7235.253
# ---
# Log-Likelihood
#      -3579.634
# ---
# Lambda:  0.1039235 std.err:  0.02154315
```

NormRegMisrepEM *Fit a Linear Regression Misrepresentation Model using EM Algorithm*

Description

NormRegMisrepEM is used to fit a Linear regression model, adjusting for misrepresentation on a binary predictor. The function uses the Expectation Maximization algorithm and allows multiple additional correctly measured independent variables in the Normal regression with an identity link function that is typically used in insurance claims modeling. Standard errors of model estimates are obtained from closed form expressions of the Observed Fisher Information.

Usage

```
NormRegMisrepEM(formula, v_star, data, lambda = c(0.6,0.4),
                 epsilon = 1e-08, maxit = 10000,
                 maxrestarts = 20, verb = FALSE)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
v_star	a character specifying the name of the binary predictor that is suspected of being misrepresented.
data	a dataframe containing the variables in the model.
lambda	initial mixing proportions used to start the EM algorithm. A numeric vector of length two, with the second element being the prevalence of misrepresentation.
epsilon	tolerance for convergence. Convergence is reached when the log-likelihood increases by less than epsilon.
maxit	the maximum number of iterations the EM routine will run for.
maxrestarts	how many times the EM routine will attempt to converge. When conergence is not achieved, the EM routine restarts with new randomly selected mixing proportions.
verb	logical. If TRUE, the difference in new .vs. old log-likelihood and the current log-likelihood is printed to the console after every iteration. If TRUE, the user will also be notifed if the EM algorithm must restart with new mixing proportions.

Details

Models for NormRegMisrepEM are specified symbolically. Like the lm and glm functions, the model has the form response ~ terms, where response is the numeric response vector and terms is a series of terms which specifies a linear predictor for response.

Currently, formula specification can accommodate the following expressions:

- transformations of the response: `log(y) ~ x`
- polynomial terms: `y ~ x + I(x^2)`
- interactions: `y ~ x*z`

Including an offset term (e.g. `y ~ x + offset()`) is currently not supported.

Value

`NormRegMisrepEM` returns an object of class `"misrepEM"`.

The function `summary` is used to obtain and print a summary of the results.

An object of class `"misrepEM"` is a list containing the following 14 elements:

`y`	the response used.
`lambda`	numeric. The estimated prevalence of misrepresentation.
`params`	a numeric vector containing the estimated parameters.
`loglik`	the final maximized log-likelihood.
`posterior`	a numeric vector. The posterior probability that the *i-th* observation is not misrepresented for observations where the suspected misrepresented variable is zero, based on the last iteration of the EM algorithm. The values are not meaningful for observations where the suspected misrepresented variable is one.
`all.loglik`	a numeric vector containing the log-likelihood at every iteration.
`cov.estimates`	the inverse of the observed fisher information matrix evaluated at the maximum likelihood estimates.
`std.error`	a numeric vector containing the standard errors of regression coefficients.
`t.values`	a numeric vector containing the standardized regression coefficients.
`p.values`	a numeric vector containing the *p*-values of the regression coefficients.
`ICs`	a numeric vector of length three containing the AIC, AICc, and BIC.
`ft`	a character containing the name of the function.
`formula`	an object of class `formula` indicating the model that was fit.
`v_star_name`	a character containing the name of the binary predictor suspected of misrepresentation.

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Akakpo, R. M., Xia, M., & Polansky, A. M. (2019). Frequentist inference in insurance ratemaking models adjusting for misrepresentation. *ASTIN Bulletin: The Journal of the IAA, 49*(1), 117-146.

Xia, M., Hua, L., & Vadnais, G. (2018). Embedded predictive analysis of misrepresentation risk in GLM ratemaking models. *Variance, 12*(1), 39-58.

Examples

```
# Simulate data
n <- 1000
p0 <- 0.25

X1 <- rbinom(n, 1, 0.4)
X2 <- sample(x = c("a", "b", "c"), size = n, replace = TRUE)
X3 <- rnorm(n, 0, 1)

theta0 <- 0.3
V <- rbinom(n,1,theta0)
V_star <- V
V_star[V==1] <- rbinom(sum(V==1),1,1-p0)

a0 <- 1
a1 <- 2
a2 <- 0
a3 <- -1
a4 <- 4
a5 <- 2

mu <- rep(0, n)

for(i in 1:n){

  mu[i] <- (a0 + a1*X1 + a4*X3 + a5*V )[i]

  if(X2[i] == "a" || X2[i] == "b"){

    mu[i] <- mu[i] + a2

  }else{
    mu[i] <- mu[i] + a3
  }

}

sigma <- 0.427

Y <- rnorm(n, mu, sigma)

data <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, V_star = V_star)

# "a" is the reference
data$X2 <- as.factor(data$X2)

# Model with main effects:
norm_lm <- NormRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star,
                              v_star = "V_star", data = data)

# The prevalence of misrepresentation;
```

```
(theta0 * p0) / (1 - theta0*(1-p0)) # 0.09677419

# Parameter estimates and estimated prevalence of
# misrepresentation (lambda);
summary(norm_lm)

# Coefficients:
#              Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  1.00624    0.02834  35.50820   <2e-16 ***
# X1           1.95903    0.02825  69.35263   <2e-16 ***
# X2b          0.04106    0.03413   1.20301  0.22926
# X2c         -1.00367    0.03418 -29.36328   <2e-16 ***
# X3           4.00031    0.01366 292.75308   <2e-16 ***
# V_star       2.01422    0.02922  68.93901   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 1674.683 1674.828 1713.945
# ---
# Log-Likelihood
#      -829.3415
# ---
# Lambda:  0.11085 std.err:  0.01150365

# Fitting an interaction between X2 and X3;

a6 <- -2
a7 <- 2

for(i in 1:n){

  if(X2[i] == "c"){
    mu[i] <- mu[i] + a6*X3[i]
  }else{
    if(X2[i] =="b"){
      mu[i] <- mu[i] + a7*X3[i]
    }
  }
}

Y <- rnorm(n, mu, sigma)

data$Y <- Y

norm_lm <- NormRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3,
                              v_star = "V_star", data = data)

summary(norm_lm)

# Coefficients:
#              Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.94905    0.02866  33.11281   <2e-16 ***
```

```
# X1           2.04258    0.02876  71.02223   <2e-16 ***
# X2b          0.00204    0.03463   0.05880  0.95313
# X2c         -0.97738    0.03469 -28.17313   <2e-16 ***
# X3           3.97014    0.02341 169.61108   <2e-16 ***
# V_star       2.01894    0.02967  68.04780   <2e-16 ***
# X2b:X3       2.00436    0.03459  57.95430   <2e-16 ***
# X2c:X3      -1.97573    0.03431 -57.59168   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 1668.925 1669.148 1718.003
# ---
# Log-Likelihood
#      -824.4626
# ---
# Lambda:  0.1055629 std.err:  0.01134299

# Model fitting with a polynomial effect;

a8 <- -0.5

mu <- mu + a8*X3^2

Y <- rnorm(n, mu, sigma)

data$Y <- Y

norm_lm <- NormRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3 + I(X3^2),
                              v_star = "V_star", data = data)

summary(norm_lm)

# Coefficients:
#             Estimate Std. Error   t value Pr(>|t|)
# (Intercept)  0.95426    0.03050  31.28435   <2e-16 ***
# X1           2.00070    0.02878  69.52668   <2e-16 ***
# X2b          0.09309    0.03480   2.67463   0.0076  **
# X2c         -0.96572    0.03455 -27.95529   <2e-16 ***
# X3           3.96765    0.02378 166.82865   <2e-16 ***
# V_star       2.00513    0.02967  67.58481   <2e-16 ***
# I(X3^2)     -0.49043    0.00983 -49.90057   <2e-16 ***
# X2b:X3       2.04613    0.03454  59.24406   <2e-16 ***
# X2c:X3      -1.97248    0.03383 -58.30381   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 1672.933 1673.200 1726.918
# ---
# Log-Likelihood
#      -825.4665
# ---
```

```
# Lambda:  0.1061873 std.err:  0.01138759
```

---

poisRegMisrepEM *Fit a Poisson Misrepresentation Model using EM Algorithm*

Description

poisRegMisrepEM is used to fit a Poisson regression model, adjusting for misrepresentation on a binary predictor. The function uses the Expectation Maximization algorithm and allows multiple additional correctly measured independent variables in the Poisson regression with a log-link function that is typically used in insurance claims modeling. Standard errors of model estimates are obtained from closed form expressions of the Observed Fisher Information.

Usage

```
poisRegMisrepEM(formula, v_star, data, lambda = c(0.6,0.4),
                 epsilon = 1e-08, maxit = 10000,
                 maxrestarts = 20, verb = FALSE)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
v_star	a character specifying the name of the binary predictor that is suspected of being misrepresented.
data	a dataframe containing the variables in the model.
lambda	initial mixing proportions used to start the EM algorithm. A numeric vector of length two, with the second element being the prevalence of misrepresentation.
epsilon	tolerance for convergence. Convergence is reached when the log-likelihood increases by less than epsilon.
maxit	the maximum number of iterations the EM routine will run for.
maxrestarts	how many times the EM routine will attempt to converge. When conergence is not achieved, the EM routine restarts with new randomly selected mixing proportions.
verb	logical. If TRUE, the difference in new .vs. old log-likelihood and the current log-likelihood is printed to the console after every iteration. If TRUE, the user will also be notifed if the EM algorithm must restart with new mixing proportions.

Details

Models for poisRegMisrepEM are specified symbolically. Like the lm and glm functions, the model has the form response ~ terms, where response is the numeric response vector and terms is a series of terms which specifies a linear predictor for response.

Currently, formula specification can accommodate the following expressions:

- transformations of the response: log(y) ~ x
- polynomial terms: y ~ x + I(x^2)
- interactions: y ~ x*z

Including an offset term (e.g. y ~ x + offset()) is currently not supported.

Value

poisRegMisrepEM returns an object of class "misrepEM".

The function summary is used to obtain and print a summary of the results.

An object of class "misrepEM" is a list containing the following 14 elements:

y	the response used.
lambda	numeric. The estimated prevalence of misrepresentation.
params	a numeric vector containing the estimated parameters.
loglik	the final maximized log-likelihood.
posterior	a numeric vector. The posterior probability that the *i-th* observation is not misrepresented for observations where the suspected misrepresented variable is zero, based on the last iteration of the EM algorithm. The values are not meaningful for observations where the suspected misrepresented variable is one.
all.loglik	a numeric vector containing the log-likelihood at every iteration.
cov.estimates	the inverse of the observed fisher information matrix evaluated at the maximum likelihood estimates.
std.error	a numeric vector containing the standard errors of regression coefficients.
z.values	a numeric vector containing the standardized regression coefficients.
p.values	a numeric vector containing the *p*-values of the regression coefficients.
ICs	a numeric vector of length three containing the AIC, AICc, and BIC.
ft	a character containing the name of the function.
formula	an object of class formula indicating the model that was fit.
v_star_name	a character containing the name of the binary predictor suspected of misrepresentation.

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Akakpo, R. M., Xia, M., & Polansky, A. M. (2019). Frequentist inference in insurance ratemaking models adjusting for misrepresentation. *ASTIN Bulletin: The Journal of the IAA, 49*(1), 117-146.

Xia, M., Hua, L., & Vadnais, G. (2018). Embedded predictive analysis of misrepresentation risk in GLM ratemaking models. *Variance, 12*(1), 39-58.

Examples

```
set.seed(314159)

# Simulate data
n <- 1000
p0 <- 0.25

X1 <- rbinom(n, 1, 0.4)
X2 <- sample(x = c("a", "b", "c"), size = n, replace = TRUE)
X3 <- rnorm(n, 0, 1)

theta0 <- 0.3
V <- rbinom(n,1,theta0)
V_star <- V
V_star[V==1] <- rbinom(sum(V==1),1,1-p0)

a0 <- 1
a1 <- 0.5
a2 <- 0
a3 <- -1
a4 <- 2
a5 <- 1

mu <- rep(0, n)

for(i in 1:n){

  mu[i] <- exp(a0 + a1*X1 + a4*X3 + a5*V )[i]

  if(X2[i] == "a" || X2[i] == "b"){

    mu[i] <- mu[i]*exp(a2)

  }else{
    mu[i] <- mu[i]*exp(a3)
  }

}

Y <- rpois(n, mu)

data <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, V_star = V_star)

# "a" is the reference
data$X2 <- as.factor(data$X2)

# Model with main effects:
pois_mod <- poisRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star,
                            v_star = "V_star", data = data)
```

```
# The prevalence of misrepresentation;
(theta0 * p0) / (1 - theta0*(1-p0)) # 0.09677419

# Parameter estimates and estimated prevalence of
# misrepresentation (lambda);
summary(pois_mod)

# Coefficients:
#             Estimate Std. Error   z value Pr(>|z|)
# (Intercept)  1.03519    0.02238  46.25615   <2e-16 ***
# X1           0.49875    0.01297  38.45157   <2e-16 ***
# X2b         -0.00007    0.01324  -0.00500  0.99601
# X2c         -0.98438    0.01926 -51.10084   <2e-16 ***
# X3           1.97794    0.00878 225.20267   <2e-16 ***
# V_star       0.99484    0.01290  77.14885   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 4170.836 4170.949 4205.190
# ---
# Log-Likelihood
#      -2078.418
# ---
# Lambda:  0.1039615 std.err:  0.01613403

# Fitting an interaction between X2 and X3;

a6 <- -0.5
a7 <- -0.5

for(i in 1:n){

  if(X2[i] == "c"){
    mu[i] <- mu[i]*exp(a6*X3[i])
  }else{
    if(X2[i] =="b"){
      mu[i] <- mu[i]*exp(a7*X3[i])
    }
  }
}

Y <- rpois(n, mu)

data$Y <- Y

pois_mod <- poisRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3,
                            v_star = "V_star", data = data)

summary(pois_mod)

# Coefficients:
#             Estimate Std. Error   z value Pr(>|z|)
```

```
# (Intercept)  0.98723    0.02917  33.84255   <2e-16 ***
# X1           0.50135    0.01540  32.56094   <2e-16 ***
# X2b         -0.03643    0.03655  -0.99648  0.31902
# X2c         -1.02315    0.05170 -19.79103   <2e-16 ***
# X3           1.99527    0.01319 151.22592   <2e-16 ***
# V_star       1.00917    0.01531  65.93335   <2e-16 ***
# X2b:X3      -0.47260    0.02137 -22.11569   <2e-16 ***
# X2c:X3      -0.49639    0.03018 -16.44530   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 4096.533 4096.714 4140.702
# ---
# Log-Likelihood
#      -2039.266
# ---
# Lambda:  0.1072814 std.err:  0.0162925

# Model fitting with a polynomial effect;

a8 <- -1

mu <- mu*exp(a8*X3^2)

Y <- rpois(n, mu)

data$Y <- Y

pois_mod <- poisRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star + X2*X3 + I(X3^2),
                            v_star = "V_star", data = data)

summary(pois_mod)

# Coefficients:
#             Estimate Std. Error   z value Pr(>|z|)
# (Intercept)  1.03291    0.04647  22.22701   <2e-16 ***
# X1           0.43783    0.03453  12.68058   <2e-16 ***
# X2b         -0.08042    0.05600  -1.43609  0.15098
# X2c         -1.02676    0.07523 -13.64912   <2e-16 ***
# X3           2.03183    0.06317  32.16597   <2e-16 ***
# V_star       0.98563    0.03415  28.86175   <2e-16 ***
# I(X3^2)     -0.99795    0.03529 -28.27715   <2e-16 ***
# X2b:X3      -0.45828    0.06499  -7.05189   <2e-16 ***
# X2c:X3      -0.47648    0.08912  -5.34623   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 3269.698 3269.920 3318.775
# ---
# Log-Likelihood
#      -1624.849
```

```
# ---
# Lambda:  0.108672 std.err:  0.02181499
```

`predict.misrepEM` *Predict method for 'misrepEM' Model Fits*

Description

Predicted values based on a fitted 'misrepEM' model object.

Usage

```
## S3 method for class 'misrepEM'
predict(object, newdata, ...)
```

Arguments

`object`	a fit from one of `gammaRegMisrepEM`, `LnRegMisrepEM`, `NormRegMisrepEM`, `nbRegMisrepEM`, or `poisRegMisrepEM`.
`newdata`	a data frame containing predictors that are to be used to make predictions of the response.
`...`	currently not used.

Details

Currently, only predictions made on the scale of the response variable are supported.

Incomplete cases are automatically dropped, and predictions are made only on complete cases.

Value

`predict.misrepEM` returns a numeric vector of predictions.

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Akakpo, R. M., Xia, M., & Polansky, A. M. (2019). Frequentist inference in insurance ratemaking models adjusting for misrepresentation. *ASTIN Bulletin: The Journal of the IAA, 49*(1), 117-146.

Xia, M., Hua, L., & Vadnais, G. (2018). Embedded predictive analysis of misrepresentation risk in GLM ratemaking models. *Variance, 12*(1), 39-58.

Examples

```
# Simulate data
n <- 2000
p0 <- 0.25

X1 <- rbinom(n, 1, 0.4)
X2 <- rnorm(n, 0, 1)
X3 <- rbeta(n, 2, 1)

theta0 <- 0.3
V <- rbinom(n,1,theta0)
V_star <- V
V_star[V==1] <- rbinom(sum(V==1),1,1-p0)

a0 <- 1
a1 <- 2
a2 <- 0
a3 <- 4
a4 <- 2

mu <- exp(a0 + a1*X1 + a2*X2 + a3*X3 + a4*V)

phi <- 0.2
alpha0 <- 1/phi
beta <- 1/mu/phi
Y <- rgamma(n, alpha0, beta)

data <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, V_star = V_star)

# Split data into training and testing sets
train <- data[1:1800,]
test  <- data[1801:2000,]

gamma_fit <- gammaRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star,
                              v_star = "V_star", data = train)

# Predict on test set;
preds <- predict(gamma_fit, newdata = test)
```

`summary.misrepEM` *Summarize a 'misrepEM' Model Fit*

Description

`summary` method for class 'misrepEM'.

Usage

```
## S3 method for class 'misrepEM'
summary(object, ...)

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

Arguments

`object`	an object of class "`misrepEM`", usually resulting from a call to one of `gammaRegMisrepEM`, `LnRegMisrepEM`, `NormRegMisrepEM`, `nbRegMisrepEM` or `poisRegMisrepEM`.
`x`	an object of class "`summary.misrepEM`", usually resulting from a call to `summary.misrepEM`.
`...`	currently not used.

Value

`summary.misrepEM` returns an object of class "`summary.misrepEM`", a list of length 5 with the following components:

`coefficients`	a `data.frame` of coefficients, standard errors, standardized coefficients, two-tailed p-values corresponding to the standardized coefficient based on a Student-t or Normal reference distribution, and 'significance stars.'
`ICs`	a named numeric vector of length three, containing the Akaike Information Criterion (AIC), the corrected AIC (AICc) and the Bayesian Information Criterion (BIC).
`loglik`	numeric. The log-likelihood of the fitted `misrepEM` model.
`lambda`	numeric. The estimated prevalence of misrepresentation.
`lambda_stderror`	numeric. The standard error of the estimated prevalence of misrepresentation.

References

Xia, Michelle, Rexford Akakpo, and Matthew Albaugh. "Maximum Likelihood Approaches to Misrepresentation Models in GLM ratemaking: Model Comparisons." *Variance* 16.1 (2023).

Akakpo, R. M., Xia, M., & Polansky, A. M. (2019). Frequentist inference in insurance ratemaking models adjusting for misrepresentation. *ASTIN Bulletin: The Journal of the IAA, 49*(1), 117-146.

Xia, M., Hua, L., & Vadnais, G. (2018). Embedded predictive analysis of misrepresentation risk in GLM ratemaking models. *Variance, 12*(1), 39-58.

Examples

```
# Simulate data
n <- 2000
p0 <- 0.25

X1 <- rbinom(n, 1, 0.4)
```

```
X2 <- rnorm(n, 0, 1)
X3 <- rbeta(n, 2, 1)

theta0 <- 0.3
V <- rbinom(n,1,theta0)
V_star <- V
V_star[V==1] <- rbinom(sum(V==1),1,1-p0)

a0 <- 1
a1 <- 2
a2 <- 0
a3 <- 4
a4 <- 2

mu <- exp(a0 + a1*X1 + a2*X2 + a3*X3 + a4*V)

phi <- 0.2
alpha0 <- 1/phi
beta <- 1/mu/phi
Y <- rgamma(n, alpha0, beta)

data <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3, V_star = V_star)

gamma_fit <- gammaRegMisrepEM(formula = Y ~ X1 + X2 + X3 + V_star,
                              v_star = "V_star", data = data)

summary(gamma_fit)

# Coefficients:
#             Estimate Std. Error  t value Pr(>|t|)
# (Intercept)  1.00137    0.03413 29.33857   <2e-16 ***
# X1           2.01388    0.02154 93.48440   <2e-16 ***
# X2          -0.00193    0.01038 -0.18589  0.85255
# X3           4.00101    0.04560 87.74528   <2e-16 ***
# V_star       2.00567    0.02240 89.54515   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
# ---
#      AIC     AICc      BIC
# 23362.50 23362.56 23401.71
# ---
# Log-Likelihood
#      -11674.25
# ---
# Lambda:  0.09635239 std.err:  0.007641834
```

Index