Package 'COMPoissonReg'

February 15, 2017

Type Package

Title Conway-Maxwell Poisson (COM-Poisson) Regression

Version 0.4.0

Date 2016-12-14

Author Kimberly Sellers <kfs7@georgetown.edu>
Thomas Lotze <thomas.lotze@thomaslotze.com>
Andrew Raim <andrew.raim@gmail.com>

Maintainer Thomas Lotze <thomas.lotze@thomaslotze.com>

Description Fit Conway-Maxwell Poisson (COM-Poisson or CMP) regression models to count data. The code provides functions for model estimation, dispersion testing, and diagnostics. Zero-inflated CMP regression is also supported.

License GPL-2 | GPL-3

LazyLoad yes

Depends stats

R topics documented:

COMPoissonReg-package *estimate parameters for COM-Poisson regression*

Description

This package offers the ability to compute the COM-Poisson parameter estimates and associated standard errors. This package also provides a hypothesis test for determining statistically significant data dispersion, and other model diagnostics.

Details

This package offers the ability to compute the COM-Poisson parameter estimates and associated standard errors for a regular regression model (via the glm.cmp function) or a zero-inflated regression model (via the glm.zicmp function).

Further, the user can perform a hypothesis test to determine the statistically significant need for using COM-Poisson regression to model the data. The test addresses the matter of statistically significant dispersion.

The main order of functions is as follows:

1. Compute Poisson estimates (using glm for Poisson regression or pscl for ZIP regression)
2. Use Poisson estimates as starting values to determine COM-Poisson estimates (using glm.cmp or glm.zicmp, respectively)
3. Compute associated standard errors (using sdev function)

From here, there are lots of ways to proceed, so order doesn't matter:

- Perform a hypothesis test to assess for statistically significant dispersion (using equitest, or non-parametrically using parametric_bootstrap)
- Compute leverage (using leverage) and deviance (using deviance)
- Predict the outcome for new examples, using predict

The package also supports fitting of the zero-inflated COM-Poisson model (ZICMP). Most of the tools available for COM-Poisson are also available for ZICMP.

Author(s)

Kimberly Sellers, Thomas Lotze, Andrew M. Raim; Maintainer: Thomas Lotze <thomas.lotze@thomaslotze.com>

References

Kimberly F. Sellers & Galit Shmueli (2010). A Flexible Regression Model for Count Data. Annals of Applied Statistics, 4(2), 943-961.

Kimberly F. Sellers and Andrew~M. Raim (2016). A Flexible Zero-Inflated Model to Address Data Dispersion. Computational Statistics and Data Analysis, 99, 68-80.

Examples

```
## load freight data
data(freight)

# Compute Standard Poisson estimates
glm_model <- glm(broken ~ transfers, data=freight,
  family=poisson, na.action=na.exclude) # beta estimates
print("The standard Poisson estimates for the beta vector are")
print(coef(glm_model))

# Compute COM-Poisson estimates (under constant dispersion model)
start.time <- Sys.time()
cmp_model = glm.cmp(formula = broken ~ transfers, data=freight)
print("The COM-Poisson estimates for the beta vector are")
print(coef(cmp_model))
print("The COM-Poisson estimate for the dispersion parameter nu is")
print(nu(cmp_model))

# Compute associated standard errors for constant COM-Poisson estimates
print("The associated standard errors for the betas in the constant dispersion case are")
print(sdev(cmp_model))

# Perform likelihood ratio test for dispersion parameter
# Test for dispersion equal or not equal to 1 (ie performing Poisson vs COM-Poisson regres
freight.test <- equitest(cmp_model)
print(sprintf("The likelihood ratio chi-squared test statistic is %0.5f
  and associated p-value (testing Poisson vs CMP regression) is %0.5f",
  freight.test$teststat, freight.test$pvalue))

# Compute constant COM-Poisson leverage
freight.lev <- leverage(cmp_model)
print("The leverage of the points is")
print(freight.lev)

# Compute constant COM-Poisson deviances
# commented-out for speed
# freight.CMPDev <- deviance(cmp_model)
# print("The approximate constant dispersion standardized CMP Deviance is")
# print(freight.CMPDev)

# Compute fitted values
freight.fitted = predict(cmp_model, newdata=freight)
print("The CMP fitted values are")
print(freight.fitted)

# Compute residual values
freight.constantCMPresids <- residuals(cmp_model)
print("The CMP residuals are")
print(freight.constantCMPresids)

# Compute MSE
freight.constantCMP.MSE <- mean(freight.constantCMPresids^2)
```

```
print("The MSE for the constant CMP regression is")
print(freight.constantCMP.MSE)

# Compute predictions on new data
new_data = data.frame(transfers=(0:10))
freight.predicted = predict(cmp_model, newdata=new_data)
plot(0:10, freight.predicted, type="l",
  xlab="number of transfers", ylab="predicted number broken")

# Compute parametric bootstrap results and use them to generate
#  0.95 confidence intervals for parameters
#  commented-out for speed
#  freight.CMPParamBoot <- parametric_bootstrap(cmp_model, n=1000)
#  print(apply(freight.CMPParamBoot,2,quantile,c(0.025,0.975)))


## load couple data
data(couple)

# Fit standard Poisson model
glm.out <- glm(UPB ~ EDUCATION + ANXIETY, data=couple, family=poisson)
print(glm.out)

# Fit ZICMP model
zicmp.out <- glm.zicmp(UPB ~ EDUCATION + ANXIETY,
  formula.nu = ~ 1,
  formula.p = ~ EDUCATION + ANXIETY,
  data=couple)
print(zicmp.out)

# Compute standard errors for estimates of coefficients
sdev(zicmp.out)

# Perform likelihood ratio test for equidispersion (H0: nu = 1 vs H1: not)
equitest(zicmp.out)

# Compute fitted values
y.hat <- predict(zicmp.out)

# Compute residuals
res.raw <- residuals(zicmp.out, type = "raw")
res.quan <- residuals(zicmp.out, type = "quantile")
plot(y.hat, res.raw)
plot(y.hat, res.quan)

# Compute predictions on new data
new_data <- data.frame(EDUCATION = round(1:20 / 20), ANXIETY = seq(-3,3, length.out = 20))
y.hat.new <- predict(zicmp.out, newdata=new_data)
plot(y.hat.new)
```

CMP Distribution *COM-Poisson Distribution*

Description

Functions for the COM-Poisson distribution.

Usage

```
dcmp(x, lambda, nu, z = NULL, log = FALSE, max = 100)
pcmp(x, lambda, nu, max = 100)
qcmp(q, lambda, nu, max = 100, log.p = FALSE)
rcmp(n, lambda, nu, max = 100)
```

Arguments

`x`	vector of quantiles.
`q`	vector of probabilities.
`n`	number of observations.
`z`	normalizing constant. Can be passed in to save computation; otherwise computed internally.
`lambda`	rate parameter.
`nu`	dispersion parameter.
`max`	maximum number to use for truncating infinite sums.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).

Value

`dcmp` gives the density, `pcmp` gives the cumulative probability, `qcmp` gives the quantile function, and `rcmp` generates random values.

Author(s)

Kimberly Sellers

References

Kimberly F. Sellers & Galit Shmueli (2010). A Flexible Regression Model for Count Data. Annals of Applied Statistics, 4(2), 943-961.

`couple.rda` *Couple dataset*

Description

A dataset investigating the impact of education level and level of anxious attachment on unwanted pursuit behaviors in the context of couple separation.

Usage

```
data(couple)
```

Format

- UPB = number of unwanted pursuit behavior perpetrations.
- EDUCATION = 1 if at least bachelor's degree; 0 otherwise.
- ANXIETY = continuous measure of anxious attachment.

References

Loeys, T., Moerkerke, B., DeSmet, O., Buysse, A., 2012. The analysis of zero-inflated count data: Beyond zero-inflated Poisson regression. British J. Math. Statist. Psych. 65 (1), 163-180.

`equitest` *Likelihood ratio test for Equidispersion*

Description

A generic function for the likelihood ratio test for equidispersion using the output of a fitted mode. The function invokes particular methods which depend on the class of the first argument.

Usage

```
 equitest(object, ...)
```

Arguments

`object`	a model object
`...`	other parameters which might be required by the model

Details

See the documentation of the particular methods for details.

Value

Returns the test statistic and p-value determined from the chi_1^2 distribution.

Author(s)

Thomas Lotze

See Also

`equitest.cmp`, `equitest.zicmp`

`freight.rda` *Freight dataset*

Description

A set of data on airfreight breakage (breakage of ampules filled with some biological substance are shipped in cartons).

Usage

```
data(freight)
```

Format

broken = number of ampules found broken upon arrival

transfers = number of times carton was transferred from one aircraft to another

References

Kutner MH, Nachtsheim CJ, Neter J (2003). Applied Linear Regression Models, Fourth Edition. McGraw-Hill.

`glm.cmp` *Functions to estimate COM-Poisson model parameters (betas and nu)*

Description

Estimates maximum likelihood estimates for betas and nu using nlminb.

Usage

```
glm.cmp(formula, initial.est = NULL, nuinit = 1, max = 100, ...)

## S3 method for class 'cmp'
AIC(object, ..., k = 2)
## S3 method for class 'cmp'
BIC(object, ...)
## S3 method for class 'cmp'
coef(object, ...)
## S3 method for class 'cmp'
deviance(object, ...)
## S3 method for class 'cmp'
equitest(object, ...)
## S3 method for class 'cmp'
leverage(object, ...)
```

```
## S3 method for class 'cmp'
logLik(object, ...)
## S3 method for class 'cmp'
nu(object, ...)
## S3 method for class 'cmp'
parametric_bootstrap(object, reps = 1000, report.period = reps + 1, ...)
## S3 method for class 'cmp'
predict(object, newdata = NULL, ...)
## S3 method for class 'cmp'
print(x, ...)
## S3 method for class 'cmp'
residuals(object, type = c("raw", "quantile"), ...)
## S3 method for class 'cmp'
sdev(object, ...)
## S3 method for class 'cmp'
summary(object, ...)
```

Arguments

`formula`	formula for the COM-Poisson model
`initial.est`	initial vector of betas, b0_1, ..., b0_p; if NULL, estimated using Poisson GLM
`nuinit`	initial value for dispersion parameter
`max`	maximum number to use for truncating infinite sums
`...`	other model parameters, such as data
`object`	object of type 'cmp'
`x`	object of type 'cmp'
`k`	Penalty per parameter to be used in AIC calculation.
`newdata`	New covariates to be used for prediction.
`type`	Type of residual to be computed.
`reps`	Number of bootstrap repetitions.
`report.period`	Report progress every `report.period` iterations.

Details

`glm.cmp` finds the maximum likelihood estimates for COM-Poisson.

Value

An object of class "cmp", from which the coefficients and other information can be computed.

Author(s)

Kimberly Sellers

References

Kimberly F. Sellers & Galit Shmueli (2010). A Flexible Regression Model for Count Data. Annals of Applied Statistics, 4(2), 943-961.

---

glm.zicmp *Functions to fit Zero-Inflated COM-Poisson model*

Description

Compute maximum likelihood estimates for parameters using optim.

Usage

```
glm.zicmp(formula.lambda, formula.nu = NULL, formula.p = NULL,
        beta.init = NULL, gamma.init = NULL, zeta.init = NULL, max = 100, ...)

## S3 method for class 'zicmp'
AIC(object, ..., k = 2)
## S3 method for class 'zicmp'
BIC(object, ...)
## S3 method for class 'zicmp'
coef(object, ...)
## S3 method for class 'zicmp'
deviance(object, ...)
## S3 method for class 'zicmp'
equitest(object, ...)
## S3 method for class 'zicmp'
leverage(object, ...)
## S3 method for class 'zicmp'
logLik(object, ...)
## S3 method for class 'zicmp'
nu(object, ...)
## S3 method for class 'zicmp'
parametric_bootstrap(object, reps = 1000, report.period = reps + 1, ...)
## S3 method for class 'zicmp'
predict(object, newdata = NULL, ...)
## S3 method for class 'zicmp'
print(x, ...)
## S3 method for class 'zicmp'
residuals(object, type = c("raw", "quantile"), ...)
## S3 method for class 'zicmp'
sdev(object, ...)
## S3 method for class 'zicmp'
summary(object, ...)
```

Arguments

`formula.lambda`
regression formula linked to `log(lambda)`

`formula.nu` regression formula linked to `log(nu)`. If NULL, is taken to be intercept only.

`formula.p` regression formula linked to `logit(p)`. If NULL, is taken to be intercept only.

`beta.init` initial values for regression coefficients of `lambda`.

`gamma.init` initial values for regression coefficients of `nu`.

`zeta.init` initial values for regression coefficients of `p`.

`max` maximum number to use for truncating infinite sums.

`...` other model parameters, such as data

`object` object of type 'zicmp'.

`x` object of type 'zicmp'.

`k` Penalty per parameter to be used in AIC calculation.

`newdata` New covariates to be used for prediction.

`type` Type of residual to be computed.

`reps` Number of bootstrap repetitions.

`report.period`
Report progress every `report.period` iterations.

Value

`glm.zicmp` produces an object of class "zicmp", from which coefficients and other information can be computed.

Author(s)

Kimberly Sellers, Andrew Raim

References

Kimberly F. Sellers & Galit Shmueli (2010). A Flexible Regression Model for Count Data. Annals of Applied Statistics, 4(2), 943-961.

Kimberly F. Sellers and Andrew~M. Raim (2016). A Flexible Zero-Inflated Model to Address Data Dispersion. Computational Statistics and Data Analysis, 99, 68-80.

`leverage` *Leverage*

Description

a generic function for the leverage of points used in various model fitting functions. The function invokes particular methods which depend on the class of the first argument.

Usage

```
leverage(object, ...)
```

Arguments

`object`	a model object
`...`	other parameters which might be required by the model

Details

See the documentation of the particular methods for details.

Value

The form of the value returned depends on the class of its argument. See the documentation of the particular methods for details of what is produced by that method.

Author(s)

Thomas Lotze

See Also

`leverage.cmp`

`nu` *Estimate for nu*

Description

a generic function for the nu estimate from the results of various model fitting functions. The function invokes particular methods which depend on the class of the first argument.

Usage

```
nu(object, ...)
```

Arguments

`object` a model object

`...` other parameters which might be required by the model

Details

See the documentation of the particular methods for details.

Value

The form of the value returned depends on the class of its argument. See the documentation of the particular methods for details of what is produced by that method.

Author(s)

Thomas Lotze

See Also

`nu.cmp`

`parametric_bootstrap` *Parametric Bootstrap*

Description

a generic function for the parametric bootstrap from the results of various model fitting functions. The function invokes particular methods which depend on the class of the first argument.

Usage

```
parametric_bootstrap(object, reps = 1000, report.period = reps+1, ...)
```

Arguments

`object` a model object

`...` other parameters which might be required by the model

`reps` Number of bootstrap repetitions.

`report.period`
Report progress every `report.period` iterations.

Details

See the documentation of the particular methods for details.

Value

The form of the value returned depends on the class of its argument. See the documentation of the particular methods for details of what is produced by that method.

Author(s)

Thomas Lotze

See Also

`parametric_bootstrap.cmp`, `parametric_bootstrap.zicmp`

`sdev` *Standard deviations*

Description

a generic function for the standard deviation estimates from the results of various model fitting functions. The function invokes particular methods which depend on the class of the first argument.

Usage

```
sdev(object, ...)
```

Arguments

`object`	a model object
`...`	other parameters which might be required by the model

Details

See the documentation of the particular methods for details.

Value

The form of the value returned depends on the class of its argument. See the documentation of the particular methods for details of what is produced by that method.

Author(s)

Thomas Lotze

See Also

`sdev.cmp`, `sdev.zicmp`

ZICMP Distribution *ZICMP Distribution*

Description

Computes the density, cumulative probability, quantiles, and random draws for the Zero-Inflated COM-Poisson distribution.

Usage

```
dzicmp(x, lambda, nu, p, z = NULL, max = 100, log = FALSE)
pzicmp(x, lambda, nu, p, max = 100)
qzicmp(q, lambda, nu, p, max = 100, log.p = FALSE)
rzicmp(n, lambda, nu, p, max = 100)
```

Arguments

`x`	vector of quantiles.
`q`	vector of probabilities.
`n`	number of observations.
`z`	normalizing constant. Can be passed in to save computation; otherwise computed internally.
`lambda`	rate parameter.
`nu`	dispersion parameter.
`p`	zero-inflation probability parameter.
`max`	maximum number to use for truncating infinite sums.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).

Value

`dzicmp` gives the density, `pzicmp` gives the cumulative probability, `qzicmp` gives the quantile value, and `rzicmp` generates random numbers.

Author(s)

Kimberly Sellers, Andrew Raim

References

Kimberly F. Sellers and Andrew~M. Raim (2016). A Flexible Zero-Inflated Model to Address Data Dispersion. Computational Statistics and Data Analysis, 99, 68-80.

Index