

COMPoissonReg Demo

Andrew Raim

November 26, 2016

COM Poisson Regression

Generate some data.

```
library(COMPoissonReg, quietly = TRUE)
set.seed(1235)

n <- 400
x <- runif(n, 1, 4)
X <- model.matrix(~ x)
beta.true <- c(1, 0.5)
lambda.true <- exp(X %*% beta.true)
nu.true <- 0.75

# compoisson::rcom doesn't seem to support vectorization
y <- numeric(n)
for (i in 1:n) {
  y[i] <- rcom(1, lambda = lambda.true[i], nu = nu.true)
}
dat <- data.frame(y = y, x = x)
```

Fit the model.

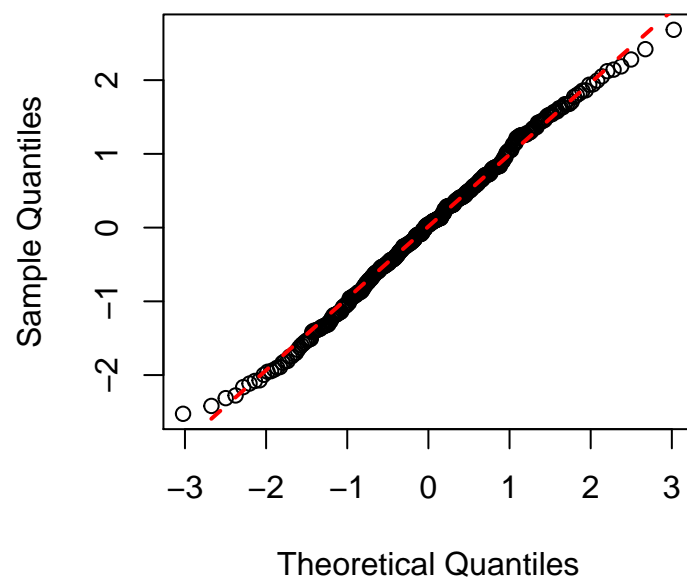
```
cmp.out <- cmp(y ~ x, data = dat)
print(cmp.out)

## Fit for CMP model
##           Estimate      SE  z.value p.value
## X:Intercept   1.0341 0.0884 11.70167      0
## X:x           0.5260 0.0389 13.52863      0
## nu            0.7841 0.0569 13.77741      0
## --
## Sample size: 400
## LogLik: -1222.7498   AIC: 2449.4996   BIC: 2457.4826
## Converged status: 0   Message: relative convergence (4)
```

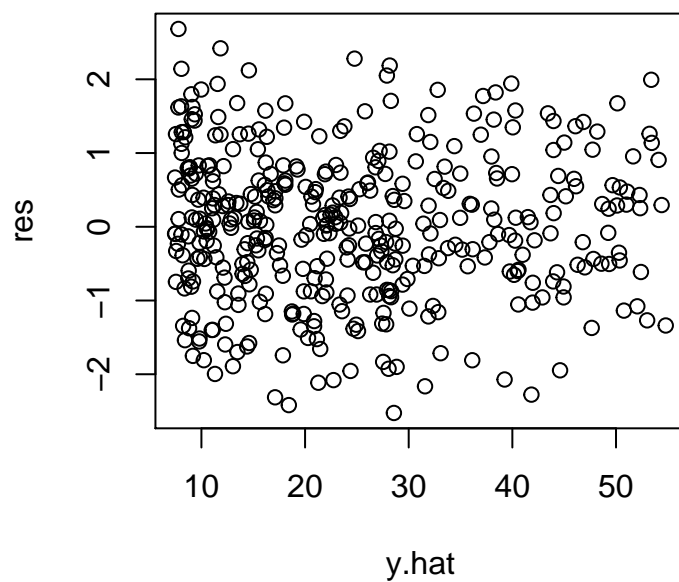
Plot the randomized quantile residuals.

```
y.hat <- predict(cmp.out)
res <- resid(cmp.out, type = "quantile")
qqnorm(res); qqline(res, lty = 2, col = "red", lwd = 2)
```

Normal Q-Q Plot



```
plot(y.hat, res)
```



Run the test for equidispersion.

```
chisq(cmp.out)
```

```
## [1] 12.2484
```

```
pval(cmp.out)
```

```
##           [,1]
```

```
## [1,] 0.0004656581
```

Zero-Inflated COM Poisson Regression

Generate some data.

```
library(COMPoissonReg, quietly = TRUE)
set.seed(1235)

n <- 400
x <- runif(n, 1, 4)
X <- model.matrix(~ x)
S <- matrix(1, n, 1)
W <- model.matrix(~ x)
beta.true <- c(1, 2)
gamma.true <- 1
zeta.true <- c(0.05, -1)
lambda.true <- exp(X %*% beta.true)
nu.true <- exp(S %*% gamma.true)
p.true <- plogis(W %*% zeta.true)

y <- r.zi.compoisson(n, lambda = lambda.true, nu = nu.true, p = p.true)
dat <- data.frame(y = y, x = x)
```

Fit the model. The package notices that we specify only an intercept for ν , and also displays an estimate for ν directly by using the appropriate transformation.

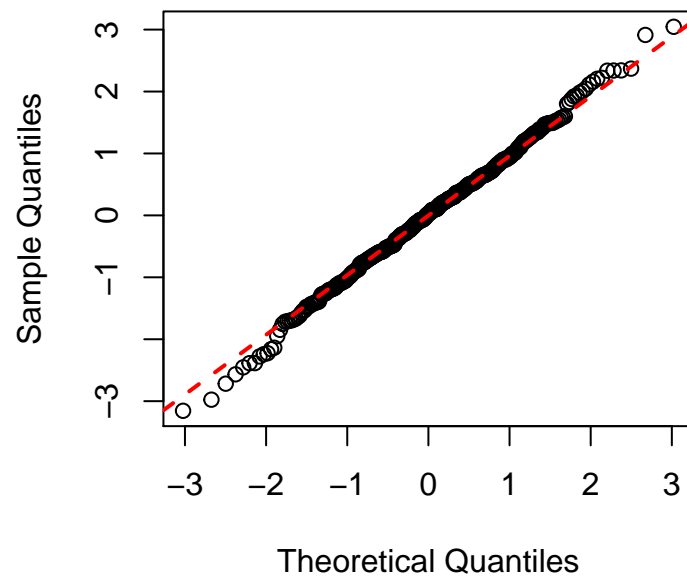
```
zicmp.out <- zicmp(y ~ x, formula.nu = ~ 1, formula.p = ~ x, data = dat)
print(zicmp.out)
```

```
## Fit for ZICMP model
##           Estimate      SE   z.value   p.value
## X:(Intercept)   0.9150 0.1370  6.678828 2.409e-11
## X:x             1.8425 0.1418 12.990885      0
## S:(Intercept)   0.9168 0.0764 12.008241      0
## W:(Intercept)   0.8417 0.5028  1.673860  0.09416
## W:x            -1.3641 0.2481 -5.498131 3.838e-08
## --
## Estimates for non-regression parameters
##      Estimate      SE   z.value   p.value
## nu    2.5014 0.191 13.09745      0
## --
## Elapsed Sec: 9.63   Sample size: 400
## LogLik: -872.9760   AIC: 1755.9520   BIC: 1775.9094
## Converged status: 0   Message: CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH
```

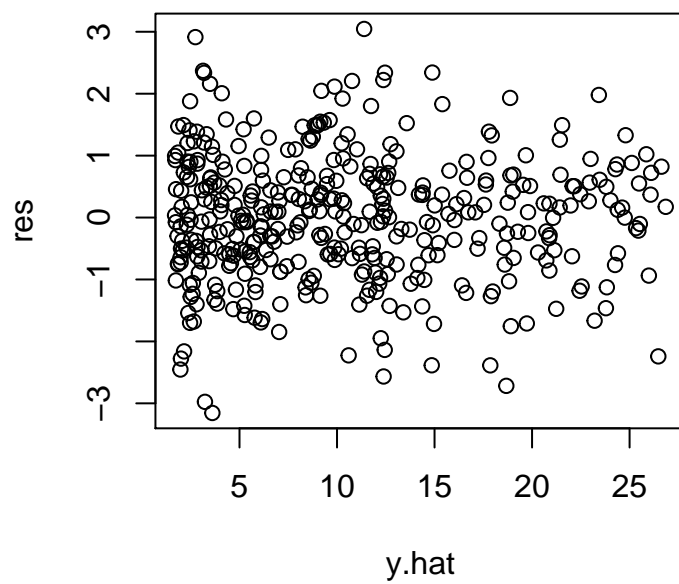
Plot the randomized quantile residuals.

```
y.hat <- predict(zicmp.out)
res <- resid(zicmp.out, type = "quantile")
qqnorm(res); qqline(res, lty = 2, col = "red", lwd = 2)
```

Normal Q-Q Plot



```
plot(y.hat, res)
```



Run the test for equidispersion.

```
chisq(zicmp.out)
```

```
## [1] 105.0832
```

```
pval(zicmp.out)
```

```
## [1] 1.171157e-24
```