Applied Econometrics with R

Chapter 5

Microeconometrics

Microeconometrics

Overview

Overview

- Many microeconometric models belong to the domain of generalized linear models (GLMs)

  Examples: probit model, Poisson regression.
- Unifying framework can be exploited in software design.
- R has a single fitting function `glm()` closely resembling `lm()`.
- Models extending GLMs are provided by R functions that analogously extend `glm()`:
  similar interfaces, return values, and associated methods.

Microeconometrics

Generalized Linear Models

Generalized linear models (GLMs)

Three aspects of linear regression model for conditionally normally distributed response y:

1. Linear predictor $\eta_i = x_i^\top \beta$ through which $\mu_i = \mathrm{E}(y_i|x_i)$ depends on $k \times 1$ vectors x_i and β.
2. Distribution of dependent variable $y_i|x_i$ is $\mathcal{N}(\mu_i, \sigma^2)$.
3. Expected response is equal to linear predictor, $\mu_i = \eta_i$.

Generalized linear models (GLMs)

Generalized linear models are defined by three elements:

1. Linear predictor $\eta_i = x_i^\top \beta$ through which $\mu_i = \mathrm{E}(y_i|x_i)$ depends on $k \times 1$ vectors x_i and β.
2. Distribution of dependent variable $y_i|x_i$ is a linear exponential family,
$$f(y; \theta, \phi) = \exp\left\{ \frac{y\theta - b(\theta)}{\phi} + c(y; \phi) \right\}$$
3. Expected response and linear predictor are related by a monotonic transformation, $g(\mu_i) = \eta_i$.
g is called the *link function* of the GLM.

   Transformation g relating original parameter μ and canonical parameter θ from exponential family representation is called *canonical link*.

Generalized linear models (GLMs)

Example 1: Poisson distribution
Probability mass function is

$$f(y; \mu) = \frac{e^{-\mu}\mu^y}{y!}, \qquad y = 0, 1, 2, \ldots$$

- Rewrite as
$$f(y; \mu) = \exp(y \log \mu - \mu - \log y!).$$
- Linear exponential family with $\theta = \log \mu$, $b(\theta) = e^\theta$, $\phi = 1$, and $c(y; \phi) = -\log y!$.
- Canonical link is logarithmic link, $\log \mu = \eta$.

Generalized linear models (GLMs)

Example 2: Bernoulli distribution
Probability mass function is

$$f(y; p) = p^y (1-p)^{1-y}, \qquad y \in \{0, 1\}.$$

- Rewrite as
$$f(y; p) = \left\{ y \log\left(\frac{p}{1-p}\right) + \log(1-p) \right\}, \quad y \in \{0, 1\}.$$
- Linear exponential family with $\theta = \log\{p/(1-p)\}$, $b(\theta) = -\log(1 + e^\theta)$, $\phi = 1$, and $c(y; \phi) = 1$.
- Canonical link: quantile function $\log\{p/(1-p)\}$ of logistic distribution (logit link).
Popular non-canonical link: quantile function Φ^{-1} of standard normal distribution (probit link).

Generalized linear models (GLMs)

Selected GLM families and their canonical (default) links:

Family	Canonical link	Name
`binomial`	$\log\{\mu/(1-\mu)\}$	`logit`
`gaussian`	μ	`identity`
`poisson`	$\log \mu$	`log`

More complete list: McCullagh and Nelder (1989).

Generalized linear models (GLMs)

- Built-in distributional assumption, hence use method of maximum likelihood (ML).
- Standard algorithm is iterative weighted least squares (IWLS) – Fisher scoring algorithm adapted for GLMs.
- Analogies with linear model suggest that fitting function could look almost like fitting function for linear models.
- In R, fitting function for GLMs is `glm()`:
  - Syntax closely resembles syntax of `lm()`.
  - Familiar arguments `formula`, `data`, `weights`, and `subset`.
  - Extra arguments for selecting response distribution and link function.
- Extractor functions known from linear models have methods for objects of class “`glm`”.

Microeconometrics

Binary Dependent Variables

Binary dependent variables

Model is

$$\mathrm{E}(y_i|x_i) = p_i = F(x_i^\top \beta), \quad i = 1, \dots, n.$$

F equal to CDF of

- standard normal distribution yields probit model.
- logistic distribution yields logit model.

Fitting logit or probit models uses `glm()` with appropriate `family` argument (including specification of `link`).

For Bernoulli outcomes

- `family` is `binomial`,
- `link` is either `link = "logit"` (default) or `link = "probit"`. Further link functions available, but not commonly used in econometrics.

Binary dependent variables

Example: Female labor force participation for 872 women from Switzerland (Gerfin, *JAE* 1996).

Dependent variable is `participation`, regressors are

- `income` – nonlabor income (in logs)
- `education` – years of formal education
- `age` – age in decades
- `youngkids` / `oldkids` – numbers of younger / older children
- `foreign` – factor indicating citizenship

Toy example of probit regression is

```
R> data("SwissLabor", package = "AER")
R> swiss_probit_ex <- glm(participation ~ age,
+    data = SwissLabor, family = binomial(link = "probit"))
```


Binary dependent variables

```
R> summary(swiss_probit_ex)
Call:
glm(formula = participation ~ age,
  family = binomial(link = "probit"), data = SwissLabor)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.260  -1.116  -0.979   1.226   1.414

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.3438     0.1672    2.06   0.0398
age          -0.1116     0.0406   -2.75   0.0059

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1203.2  on 871  degrees of freedom
Residual deviance: 1195.7  on 870  degrees of freedom
AIC: 1200

Number of Fisher Scoring iterations: 4
```


Binary dependent variables

Gerfin's model is

```
R> swiss_probit <- glm(participation ~ . + I(age^2),
+    data = SwissLabor, family = binomial(link = "probit"))
R> coeftest(swiss_probit)

z test of coefficients:

            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.7491     1.4069    2.66   0.0077
income       -0.6669     0.1320   -5.05  4.3e-07
age           2.0753     0.4054    5.12  3.1e-07
education     0.0192     0.0179    1.07   0.2843
youngkids    -0.7145     0.1004   -7.12  1.1e-12
oldkids      -0.1470     0.0509   -2.89   0.0039
foreignyes    0.7144     0.1213    5.89  3.9e-09
I(age^2)     -0.2943     0.0499   -5.89  3.8e-09
```


Visualization

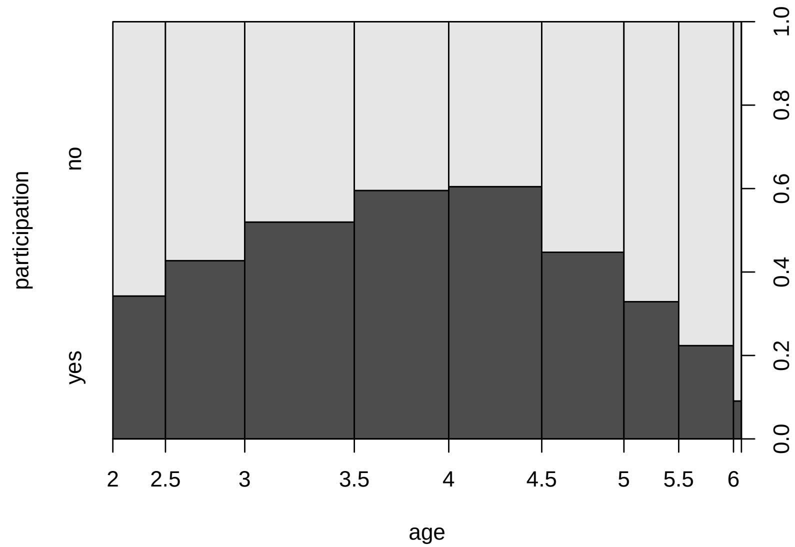
Use spinogram:

- Groups regressor age into intervals (as in histogram).
- Produces spine plot for resulting proportions of `participation` within age groups.

In R:

```
R> plot(participation ~ age, data = SwissLabor, ylevels = 2:1)
```

Visualization



Effects

Effects in probit model vary with regressors:

$$\frac{\partial \mathrm{E}(y_i|x_i)}{\partial x_{ij}} = \frac{\partial \Phi(x_i^\top \beta)}{\partial x_{ij}} = \phi(x_i^\top \beta) \cdot \beta_j$$

Researchers often report average marginal effects.

Several versions of such averages:

- Average of the sample marginal effects

$$\frac{1}{n} \sum_{i=1}^{n} \phi(x_i^\top \hat\beta) \cdot \hat\beta_j$$

- Effect evaluated at average regressor

Effects

Version 1: Average of sample marginal effects is

```
R> fav <- mean(dnorm(predict(swiss_probit, type = "link")))
R> fav * coef(swiss_probit)
```

```
(Intercept)      income         age   education   youngkids
   1.241930   -0.220932    0.687466    0.006359   -0.236682
    oldkids  foreignyes    I(age^2)
  -0.048690    0.236644   -0.097505
```


Effects

Version 2: Effect evaluated at average regressors is

```
R> av <- colMeans(SwissLabor[, -c(1, 7)])
R> av <- data.frame(rbind(swiss = av, foreign = av),
+    foreign = factor(c("no", "yes")))
R> av <- predict(swiss_probit, newdata = av, type = "link")
R> av <- dnorm(av)
```

giving

```
R> av["swiss"] * coef(swiss_probit)[-7]
```

```
(Intercept)      income         age   education   youngkids
   1.495137   -0.265976    0.827628    0.007655   -0.284938
    oldkids    I(age^2)
  -0.058617   -0.117384
```

```
R> av["foreign"] * coef(swiss_probit)[-7]
```

```
(Intercept)      income         age   education   youngkids
   1.136517   -0.202180    0.629115    0.005819   -0.216593
    oldkids    I(age^2)
  -0.044557   -0.089229
```

Thus all effects are smaller in absolute size for foreigners.

Goodness of fit and prediction

McFadden's pseudo-R^2 is

$$R^2 = 1 - \frac{\ell(\hat{\beta})}{\ell(\bar{y})},$$

with $\ell(\hat{\beta})$ log-likelihood for fitted model and $\ell(\bar{y})$ log-likelihood for model with only constant term.

In R:

- Compute null model.
- Extract `logLik()` values for the two models.

```
R> swiss_probit0 <- update(swiss_probit, formula = . ~ 1)
R> 1 - as.vector(logLik(swiss_probit)/logLik(swiss_probit0))

[1] 0.1546
```


Goodness of fit and prediction

Confusion matrix needs prediction for GLMs.
Several types of predictions:

- `"link"` (default) – on scale of linear predictors.
- `"response"` – on scale of mean of response.

To obtain confusion matrix:

- Round predicted probabilities.
- Tabulate result against actual values of `participation`.

```
R> table(true = SwissLabor$participation,
+    pred = round(fitted(swiss_probit)))

     pred
true    0   1
  no  337 134
  yes 146 255
```

Thus 67.89% correctly classified and 32.11% misclassified observations.

Goodness of fit and prediction

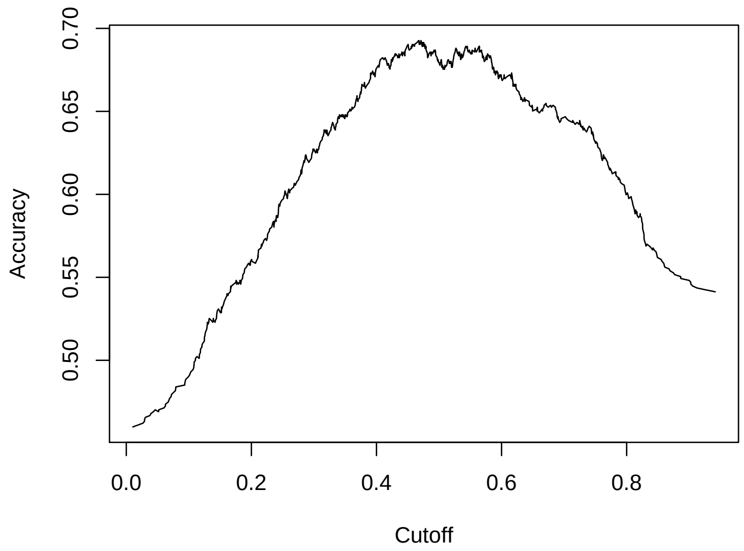
Accuracy:

- Confusion matrix uses arbitrarily chosen cutoff 0.5 for predicted probabilities.
- To avoid choosing particular cutoff:
  Evaluate performance for every conceivable cutoff; e.g., using *accuracy* of the model – proportion of correctly classified observations.
- Package **ROCR** provides necessary tools.

In R:

```
R> library("ROCR")
R> pred <- prediction(fitted(swiss_probit),
+    SwissLabor$participation)
R> plot(performance(pred, "acc"))
```


Goodness of fit and prediction



Goodness of fit and prediction

Receiver operating characteristic (ROC) curve.
Plots, for every cutoff $c \in [0, 1]$,

- **true positive rate** TPR(c)
  Number of women participating in labor force that are classified as participating compared with total number of women participating.

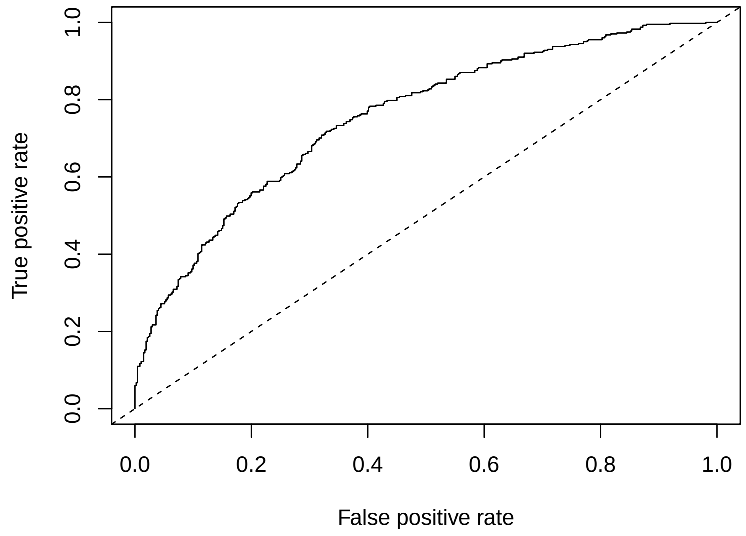
against

- **false positive rate** FPR(c)
  Number of women not participating in labor force that are classified as participating compared with total number of women not participating.

In R:

```
R> plot(performance(pred, "tpr", "fpr"))
R> abline(0, 1, lty = 2)
```


Goodness of fit and prediction



Residuals and diagnostics

`residuals()` method for "`glm`" objects provides

- Deviance residuals (signed contributions to overall deviance).
- Pearson residuals (often called standardized residuals in econometrics).
- In addition, have working, raw (or response), and partial residuals.

Sums of squares:

```
R> deviance(swiss_probit)
[1] 1017
R> sum(residuals(swiss_probit, type = "deviance")^2)
[1] 1017
R> sum(residuals(swiss_probit, type = "pearson")^2)
[1] 866.5
```


Residuals and diagnostics

Further remarks:

- Analysis of deviance via `anova()` method for "`glm`" objects.
- Sandwich estimates of covariance matrix available via `coeftest()` in the usual manner.
  Warning: Not recommended for binary regressions – variance and regression equation are either both correctly specified or not!

(Quasi-)complete separation

Example: from Maddala (2001), *Introduction to Econometrics*, 3e

Consider indicator of the incidence of `executions` in USA during 1946–1950. Observations are 44 US states.
Regressors are

- `rate` – Murder rate per 100,000 (FBI estimate, 1950).
- `convictions` – Number of convictions divided by number of murders in 1950.
- `time` – Median time served (in months) of convicted murderers released in 1951.
- `income` – Median family income in 1949 (in 1,000 USD).
- `lfp` – Labor force participation rate in 1950 (in percent).
- `noncauc` – Proportion of non-Caucasian population in 1950.
- `southern` – Factor indicating region.

(Quasi-)complete separation

```
R> data("MurderRates")
R> murder_logit <- glm(I(executions > 0) ~ time + income +
+    noncauc + lfp + southern, data = MurderRates,
+    family = binomial)

Warning message:
fitted probabilities numerically 0 or 1 occurred in:
glm.fit(x = X, y = Y, weights = weights, start = start,
```

```
R> coeftest(murder_logit)

z test of coefficients:

            Estimate Std. Error z value Pr(>|z|)
(Intercept)  10.9933    20.7734    0.53    0.597
time          0.0194     0.0104    1.87    0.062
income       10.6101     5.6541    1.88    0.061
noncauc      70.9879    36.4118    1.95    0.051
lfp          -0.6676     0.4767   -1.40    0.161
southernyes  17.3313  2872.1707    0.01    0.995
```


(Quasi-)complete separation

```
R> murder_logit2 <- glm(I(executions > 0) ~ time + income +
+    noncauc + lfp + southern, data = MurderRates,
+    family = binomial, control = list(epsilon = 1e-15,
+    maxit = 50, trace = FALSE))

Warning message:
fitted probabilities numerically 0 or 1 occurred in:
glm.fit(x = X, y = Y, weights = weights, start = start,
```

```
R> coeftest(murder_logit2)

z test of coefficients:

            Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.10e+01   2.08e+01    0.53    0.597
time        1.94e-02   1.04e-02    1.87    0.062
income      1.06e+01   5.65e+00    1.88    0.061
noncauc     7.10e+01   3.64e+01    1.95    0.051
lfp        -6.68e-01   4.77e-01   -1.40    0.161
southernyes 3.33e+01   1.73e+07    0.00    1.000
```


(Quasi-)complete separation

Phenomenon:

- Warning message: some fitted probabilities are numerically identical to zero or one, standard error of `southern` is large.
- After changing controls: warning does not go away, coefficient doubles, 6,000-fold increase of standard error.

Explanation:

- Data exhibit quasi-complete separation.
- MLE does not exist (likelihood bounded but no interior maximum).

```
R> table(I(MurderRates$executions > 0), MurderRates$southern)

        no yes
  FALSE  9   0
  TRUE  20  15
```

What to do? Depends on context!

Microeconometrics

Regression Models for Count Data

Regression Models for Count Data

Example: `RecreationDemand` data

Regress `trips` – number of recreational boating trips to Lake Somerville, TX, in 1980 – on

- `quality` – Facility's subjective quality ranking (scale of 1 to 5).
- `ski` – Water-skiing at the lake? (Factor)
- `income` – Annual household income (in 1,000 USD).
- `userfee` – Annual user fee paid at Lake Somerville? (Factor)
- `costC` – Expenditure when visiting Lake Conroe.
- `costS` – Expenditure when visiting Lake Somerville.
- `costH` – Expenditure when visiting Lake Houston.

Regression Models for Count Data

Standard model: Poisson regression with log link

$$\mathsf{E}(y_i|x_i) = \mu_i = \exp(x_i^\top \beta).$$

In R:

```
R> data("RecreationDemand")
R> rd_pois <- glm(trips ~ ., data = RecreationDemand,
+    family = poisson)
R> coeftest(rd_pois)
z test of coefficients:

            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26499    0.09372    2.83   0.0047
quality      0.47173    0.01709   27.60  < 2e-16
skiyes       0.41821    0.05719    7.31  2.6e-13
income      -0.11132    0.01959   -5.68  1.3e-08
userfeeyes   0.89817    0.07899   11.37  < 2e-16
costC       -0.00343    0.00312   -1.10   0.2713
costS       -0.04254    0.00167  -25.47  < 2e-16
costH        0.03613    0.00271   13.34  < 2e-16
```


Dealing with overdispersion

Poisson distribution has $\mathsf{E}(y) = \mathsf{Var}(y)$ – equidispersion.
In economics typically $\mathsf{E}(y) < \mathsf{Var}(y)$ – overdispersion (OD).

Test for OD: use alternative hypothesis (Cameron and Trivedi 1990)

$$\mathsf{Var}(y_i|x_i) = \mu_i + \alpha \cdot h(\mu_i), \qquad h(\mu) \geq 0$$

$\alpha > 0$ overdispersion and $\alpha < 0$ underdispersion.

- Estimate α by auxiliary OLS regression.
- Test via corresponding t statistic.

Common specifications are

- $h(\mu) = \mu^2$ (NB2)
  "negative binomial model with quadratic variance function"
- $h(\mu) = \mu$ (NB1)
  "negative binomial model with linear variance function"

Dealing with overdispersion

```
R> dispersiontest(rd_pois)

        Overdispersion test

data:  rd_pois
z = 2.4, p-value = 0.008
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
     6.566
```

and

```
R> dispersiontest(rd_pois, trafo = 2)

        Overdispersion test

data:  rd_pois
z = 2.9, p-value = 0.002
alternative hypothesis: true alpha is greater than 0
sample estimates:
alpha
1.316
```


Dealing with overdispersion

In statistical literature, reparameterization of NB1 with

$$\text{Var}(y_i|x_i) \quad = \quad (1+\alpha)\cdot\mu_i = \text{dispersion}\cdot\mu_i$$

is called *quasi-Poisson model with dispersion parameter*.

`glm()` also offers quasi-Poisson model:

```
R> rd_qpois <- glm(trips ~ ., data = RecreationDemand,
+    family = quasipoisson)
```


Dealing with overdispersion

More flexible distribution is *negative binomial* with probability density function

$$f(y;\mu,\theta) = \frac{\Gamma(\theta+y)}{\Gamma(\theta)y!}\frac{\mu^y\theta^\theta}{(\mu+\theta)^{y+\theta}}, \quad y = 0,1,2,\dots,\mu>0,\theta>0.$$

- Variance is
$$\text{Var}(y;\mu,\theta) = \mu + \frac{1}{\theta}\mu^2$$
This is NB2 with $h(\mu)=\mu^2$ and $\alpha = 1/\theta$.
- For θ known, negative binomial is exponential family.
- Poisson distribution with parameter μ for $\theta\to\infty$.
- Geometric distribution for $\theta = 1$.

Dealing with overdispersion

```
R> library("MASS")
R> rd_nb <- glm.nb(trips ~ ., data = RecreationDemand)
R> coeftest(rd_nb)

z test of coefficients:

            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.12194    0.21430   -5.24  1.6e-07
quality      0.72200    0.04012   18.00  < 2e-16
skiyes       0.61214    0.15030    4.07  4.6e-05
income      -0.02606    0.04245   -0.61    0.539
userfeeyes   0.66917    0.35302    1.90    0.058
costC        0.04801    0.00918    5.23  1.7e-07
costS       -0.09269    0.00665  -13.93  < 2e-16
costH        0.03884    0.00775    5.01  5.4e-07

R> logLik(rd_nb)

'log Lik.' -825.6 (df=9)
```

Shape parameter is $\hat{\theta} = 0.7293$.

Robust standard errors

Further way to deal with OD:

- Use Poisson estimates of the mean function.
- Adjust standard errors via sandwich formula (“Huber-White standard errors”).

Compare Poisson with Huber-White standard errors:

```
R> round(sqrt(rbind(diag(vcov(rd_pois)),
+    diag(sandwich(rd_pois)))), digits = 3)

     (Intercept) quality skiyes income userfeeyes costC costS
[1,]       0.094   0.017  0.057   0.02      0.079 0.003 0.002
[2,]       0.432   0.049  0.194   0.05      0.247 0.015 0.012
     costH
[1,] 0.003
[2,] 0.009
```


Robust standard errors

Regression output with robust standard errors via `coeftest()`:

```
R> coeftest(rd_pois, vcov = sandwich)

z test of coefficients:

             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26499    0.43248    0.61  0.54006
quality      0.47173    0.04885    9.66  < 2e-16
skiyes       0.41821    0.19387    2.16  0.03099
income      -0.11132    0.05031   -2.21  0.02691
userfeeyes   0.89817    0.24691    3.64  0.00028
costC       -0.00343    0.01470   -0.23  0.81549
costS       -0.04254    0.01173   -3.62  0.00029
costH        0.03613    0.00939    3.85  0.00012
```

Can also have OPG standard errors using `vcovOPG()`.

Zero-inflated Poisson and negative binomial models

Typical problem with count data : too many zeros

- `RecreationDemand` example has 63.28% zeros.
- Poisson regression provides only 41.96%.

Compare observed and expected counts:

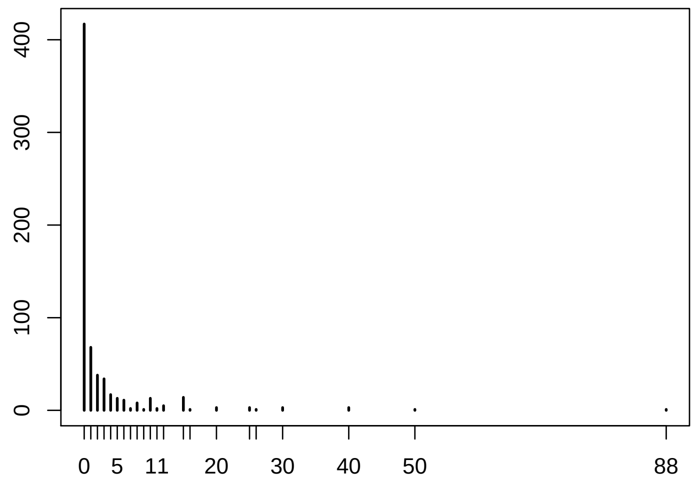
```
R> rbind(obs = table(RecreationDemand$trips)[1:10], exp = round(
+    sapply(0:9, function(x) sum(dpois(x, fitted(rd_pois))))))

      0   1  2  3  4  5  6  7  8 9
obs 417  68 38 34 17 13 11  2  8 1
exp 277 146 68 41 30 23 17 13 10 7
```

Plot marginal distribution of response:

```
R> plot(table(RecreationDemand$trips), ylab = "")
```


Zero-inflated Poisson and negative binomial models



Zero-inflated Poisson and negative binomial models

Zero-inflated Poisson (ZIP) model (Mullahy 1986, Lambert 1992)

$$f_{\text{zeroinfl}}(y) = p_i \cdot I_{\{0\}}(y) \; + \; (1 - p_i) \cdot f_{\text{count}}(y; \mu_i)$$

- Mixture with (Poisson) count component and additional point mass at zero.
- μ_i and p_i are modeled as functions of covariates.
- For count part, canonical link gives $\log(\mu_i) = x_i^\top \beta$.
- For binary part, $g(p_i) = z_i^\top \gamma$ for some quantile function g. Canonical link (logit) uses logistic distribution, probit uses standard normal.
- Sets of regressors x_i and z_i need not be identical.

Zero-inflated Poisson and negative binomial models

In R: pscl provides `zeroinfl()` for fitting zero-inflation models.

- Count component: Poisson, geometric, and negative binomial distributions, with log link.
- Binary component: all standard links, default is logit.

Example: (Cameron and Trivedi 1998)

Zero-inflated negative binomial (ZINB) for recreational trips

```
R> library("pscl")
R> rd_zinb <- zeroinfl(trips ~ . | quality + income,
+    data = RecreationDemand, dist = "negbin")
R> summary(rd_zinb)
```


Zero-inflated Poisson and negative binomial models

```
Call:
zeroinfl(formula = trips ~ . | quality + income,
  data = RecreationDemand, dist = "negbin")

Pearson residuals:
    Min      1Q  Median      3Q     Max
-1.0889 -0.2004 -0.0570 -0.0451 40.0139

Count model coefficients (negbin with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.09663    0.25668    4.27  1.9e-05
quality      0.16891    0.05303    3.19   0.0014
skiyes       0.50069    0.13449    3.72   0.0002
income      -0.06927    0.04380   -1.58   0.1138
userfeeyes   0.54279    0.28280    1.92   0.0549
costC        0.04044    0.01452    2.79   0.0053
costS       -0.06621    0.00775   -8.55  < 2e-16
```


Zero-inflated Poisson and negative binomial models

```
costH        0.02060    0.01023    2.01   0.0441
Log(theta)   0.19017    0.11299    1.68   0.0924

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    5.743      1.556    3.69  0.00022
quality       -8.307      3.682   -2.26  0.02404
income        -0.258      0.282   -0.92  0.35950

Theta = 1.209
Number of iterations in BFGS optimization: 26
Log-likelihood: -722 on 12 Df
```

Expected counts are

```
R> round(colSums(predict(rd_zinb, type = "prob")[,1:10]))
  0   1   2   3   4   5   6   7   8   9
433  47  35  27  20  16  12  10   8   7
```

Note: `predict()` method for `type = "prob"` returns matrix with vectors of expected probabilities for each observation.
Must take column sums for expected counts.

Zero-inflated Poisson and negative binomial models

Hurdle model: (Mullahy 1986)

A "two-part model" with

- binary part (given by a count distribution right-censored at $y = 1$): Is y_i equal to zero or positive? "Is the hurdle crossed?"
- count part (given by a count distribution left-truncated at $y = 1$): If $y_i > 0$, how large is y_i?

Results in

$$f_{\text{hurdle}}(y; x, z, \beta, \gamma) = \begin{cases} f_{\text{zero}}(0; z, \gamma), & \text{if } y = 0, \\ \{1 - f_{\text{zero}}(0; z, \gamma)\} \cdot f_{\text{count}}(y; x, \beta)/\{1 - f_{\text{count}}(0; x, \beta)\}, & \text{if } y > 0. \end{cases}$$

Zero-inflated Poisson and negative binomial models

In R:

- Package **pscl** provides a function `hurdle()`
- *Warning:* there are several parameterizations for binary part! In `hurdle()`, can specify either
  - count distribution right-censored at one, or
  - Bernoulli distribution distinguishing between zeros and non-zeros (equivalent to right-censored geometric distribution)

Example: (Cameron and Trivedi 1998)

Negative binomial hurdle model for recreational trips

```
R> rd_hurdle <- hurdle(trips ~ . | quality + income,
+    data = RecreationDemand, dist = "negbin")
R> summary(rd_hurdle)
```


Zero-inflated Poisson and negative binomial models

```
Call:
hurdle(formula = trips ~ . | quality + income,
  data = RecreationDemand, dist = "negbin")

Pearson residuals:
   Min     1Q Median     3Q    Max
-1.610 -0.207 -0.185 -0.164 12.111

Count model coefficients (truncated negbin with log link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.8419     0.3828    2.20   0.0278
quality       0.1717     0.0723    2.37   0.0176
skiyes        0.6224     0.1901    3.27   0.0011
income       -0.0571     0.0645   -0.88   0.3763
userfeeyes    0.5763     0.3851    1.50   0.1345
costC         0.0571     0.0217    2.63   0.0085
```


Zero-inflated Poisson and negative binomial models

```
costS        -0.0775     0.0115   -6.71  1.9e-11
costH         0.0124     0.0149    0.83   0.4064
Log(theta)   -0.5303     0.2611   -2.03   0.0423
Zero hurdle model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.7663     0.3623   -7.64  2.3e-14
quality       1.5029     0.1003   14.98  < 2e-16
income       -0.0447     0.0785   -0.57     0.57

Theta: count = 0.588
Number of iterations in BFGS optimization: 18
Log-likelihood: -765 on 12 Df
```

Expected counts are

```
R> round(colSums(predict(rd_hurdle, type = "prob")[,1:10]))
  0   1   2   3   4   5   6   7   8   9
417  74  42  27  19  14  10   8   6   5
```

Considerable improvement over Poisson specification.
More details: Zeileis, Kleiber and Jackman (*JSS* 2008).

Microeconometrics

Censored Dependent Variables

Censored Dependent Variables

Tobit model (J. Tobin, *Econometrica* 1958)

$$y_i^0 = x_i^\top \beta + \varepsilon_i, \qquad \varepsilon_i | x_i \sim \mathcal{N}(0, \sigma^2) \text{ i.i.d.},$$

$$y_i = \begin{cases} y_i^0, & y_i^0 > 0, \\ 0, & y_i^0 \le 0. \end{cases}$$

Log-likelihood is

$$\ell(\beta, \sigma^2) = \sum_{y_i > 0} \left(\log \phi\{(y_i - x_i^\top \beta)/\sigma\} - \log \sigma \right) + \sum_{y_i = 0} \log \Phi(-x_i^\top \beta / \sigma).$$

- Special case of a censored regression model.
- R package for fitting has long been available: **survival** (Therneau and Grambsch 2000).
- **AER** has convenience function `tobit()` interfacing `survreg()`.

Censored Dependent Variables

Example: “Fair’s affairs” (Fair, *JPE* 1978)
Survey on extramarital affairs conducted by *Psychology Today* (1969).
Dependent variable is `affairs` (number of extramarital affairs during past year), regressors are

- `gender` – Factor indicating gender.
- `age` – Age in years.
- `yearsmarried` – Number of years married.
- `children` – Are there children in the marriage? (factor)
- `religiousness` – Numeric variable coding religiousness (from 1 = anti to 5 = very).
- `education` – Level of education (numeric variable).
- `occupation` – Occupation (numeric variable).
- `rating` – Self rating of marriage (numeric from 1 = very unhappy to 5 = very happy).

Censored Dependent Variables

In R:

Toy example:

```
R> data("Affairs")
R> aff_tob_ex <- tobit(affairs ~ yearsmarried, data = Affairs)
```

Fair’s model:

```
R> aff_tob <- tobit(affairs ~ age + yearsmarried +
+    religiousness + occupation + rating, data = Affairs)
```

Censored Dependent Variables

```
Call:
tobit(formula = affairs ~ yearsmarried, data = Affairs)

Observations:
         Total  Left-censored     Uncensored Right-censored
           601            451            150              0

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -9.2629     1.1723   -7.90  2.8e-15
yearsmarried   0.3758     0.0899    4.18  2.9e-05
Log(scale)     2.2092     0.0680   32.47  < 2e-16

Scale: 9.11

Gaussian distribution
Number of Newton-Raphson Iterations: 3
Log-likelihood: -736 on 3 Df
Wald-statistic: 17.5 on 1 Df, p-value: 2.9e-05
```


Censored Dependent Variables

Fair's model:

```
R> coeftest(aff_tob)

z test of coefficients:

              Estimate Std. Error z value Pr(>|z|)
(Intercept)     8.1742     2.7414    2.98   0.0029
age            -0.1793     0.0791   -2.27   0.0234
yearsmarried    0.5541     0.1345    4.12  3.8e-05
religiousness  -1.6862     0.4038   -4.18  3.0e-05
occupation      0.3261     0.2544    1.28   0.2000
rating         -2.2850     0.4078   -5.60  2.1e-08
Log(scale)      2.1099     0.0671   31.44  < 2e-16
```


Censored Dependent Variables

Refitting with additional censoring from the right:

```
R> aff_tob2 <- update(aff_tob, right = 4)
R> coeftest(aff_tob2)

z test of coefficients:

              Estimate Std. Error z value Pr(>|z|)
(Intercept)     7.9010     2.8039    2.82  0.00483
age            -0.1776     0.0799   -2.22  0.02624
yearsmarried    0.5323     0.1412    3.77  0.00016
religiousness  -1.6163     0.4244   -3.81  0.00014
occupation      0.3242     0.2539    1.28  0.20162
rating         -2.2070     0.4498   -4.91  9.3e-07
Log(scale)      2.0723     0.1104   18.77  < 2e-16
```

Standard errors now somewhat larger $\to$ heavier censoring leads to loss of information.

Note: `tobit()` has argument `dist` for alternative distributions of latent variable (logistic, Weibull, . . .).

Censored Dependent Variables

Wald-type test with sandwich standard errors:

```
R> linearHypothesis(aff_tob, c("age = 0", "occupation = 0"),
+    vcov = sandwich)

Linear hypothesis test

Hypothesis:
age = 0
occupation = 0

Model 1: restricted model
Model 2: affairs ~ age + yearsmarried + religiousness +
  occupation + rating

Note: Coefficient covariance matrix supplied.

  Res.Df Df Chisq Pr(>Chisq)
1    596
2    594  2  4.91      0.086
```

Thus regressors `age` and `occupation` jointly weakly significant.

Microeconometrics

Extensions

Extensions

Further packages for microeconometrics:

- **gam** – Generalized additive models.
- **lme4** – Nonlinear random-effects models: counts, binary dependent variables, etc.
- **mgcv** – Generalized additive (mixed) models.
- **micEcon** – Demand systems, cost and production functions.
- **mlogit** – Multinomial logit models with choice-specific variables.
- **robustbase** – Robust/resistant regression for GLMs.
- **sampleSelection** – Selection models: generalized tobit, heckit.

A semiparametric binary response model

Log-likelihood of binary response model is

$$\ell(\beta) = \sum_{i=1}^{n} \left\{ y_i \log F(x_i^\top \beta) + (1 - y_i) \log\{1 - F(x_i^\top \beta)\} \right\},$$

with F CDF of logistic or Gaussian distribution.
Klein and Spady (*Econometrica* 1993) estimate F via kernel methods – a semiparametric MLE.

In R: Klein and Spady estimator available in **np**.
Need some preprocessing:

```
R> SwissLabor$partnum <- as.numeric(SwissLabor$participation) - 1
```

First compute bandwidth object:

```
R> library("np")
R> swiss_bw <- npindexbw(partnum ~ income + age + education +
+    youngkids + oldkids + foreign + I(age^2), data = SwissLabor,
+    method = "kleinspady", nmulti = 5)
```


A semiparametric binary response model

Summary of the bandwidths is

```
R> summary(swiss_bw)

Single Index Model
Regression data (872 observations, 7 variable(s)):

      income   age education youngkids oldkids foreign I(age^2)
Beta:      1 2.023   -0.1776    -3.945  0.5071   1.802  -0.4991
Bandwidth:  0.1838
Optimisation Method:  Nelder-Mead
Regression Type: Local-Constant
Bandwidth Selection Method: Klein and Spady
Formula: partnum ~ income + age + education + youngkids +
  oldkids + foreign + I(age^2)
Bandwidth Type: Fixed
Objective Function Value: 0.6154 (achieved on multistart 2)

Continuous Kernel Type: Second-Order Gaussian
No. Continuous Explanatory Vars.: 1
Estimation Time: 173.7 seconds
```

A semiparametric binary response model

Finally pass bandwidth object `swiss_bw` to `npindex()`:

```
R> swiss_ks <- npindex(bws = swiss_bw, gradients = TRUE)
R> summary(swiss_ks)
Single Index Model
Regression Data: 872 training points, in 7 variable(s)

      income   age education youngkids oldkids foreign I(age^2)
Beta:      1 2.023   -0.1776    -3.945  0.5071   1.802  -0.4991
Bandwidth: 0.1838
Kernel Regression Estimator: Local-Constant

Confusion Matrix
      Predicted
Actual   0   1
     0 322 149
     1 130 271

Overall Correct Classification Ratio:  0.68
Correct Classification Ratio By Outcome:
     0      1
...
```


A semiparametric binary response model

Compare confusion matrix with confusion matrix of original probit:

```
R> table(Actual = SwissLabor$participation, Predicted =
+    round(predict(swiss_probit, type = "response")))
      Predicted
Actual   0   1
   no  337 134
   yes 146 255
```

Thus semiparametric model has slightly better (in-sample) performance.

Warning: these methods are time-consuming!

Multinomial responses

Describe $P(y_i = j) = p_{ij}$ via, e.g.,

$$\eta_{ij} = \log \frac{p_{ij}}{p_{i1}}, \quad j = 2, \dots, m$$

Here category 1 is reference category (needed for identification).

Variants:

- Individual-specific covariates ($\eta_{ij} = x_i^\top \beta_j$)
- Outcome-specific covariates ($\eta_{ij} = z_{ij}^\top \gamma$, “conditional logit”)
- Individual- and outcome-specific covariates (“mixed logit”)

In R:

- Function `multinom()` from **nnet** fits multinomial logits with individual-specific covariates.
- Function `mlogit()` from **mlogit** also fits mixed logits.

Here we only use `multinom()`.

Multinomial responses

Example: (from Heij, de Boer, Franses, Kloek, and van Dijk 2004)
Regress `job` – ordered factor indicating job category, with levels `"custodial"`, `"admin"` and `"manage"` – on regressors

- `education` – Education in years.
- `gender` – Factor indicating gender.
- `minority` – Factor. Is the employee member of a minority?

Multinomial responses

First overview: generate table of conditional proportions via

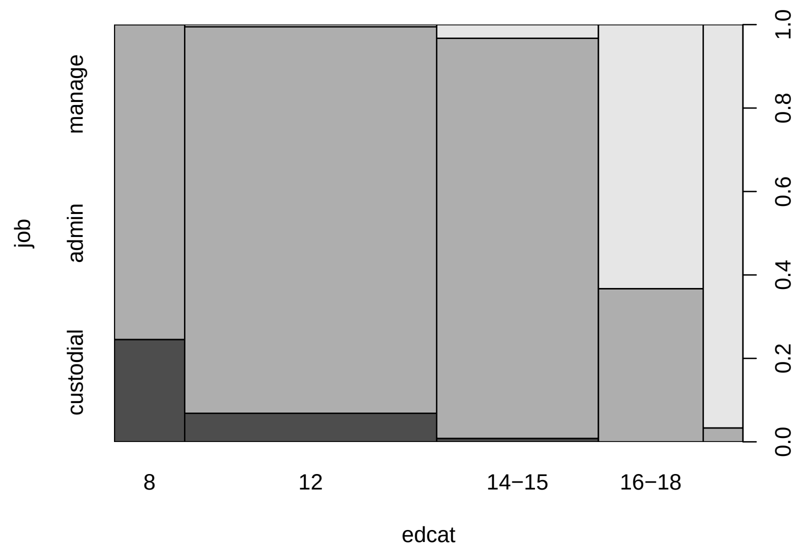
```
R> data("BankWages")
R> edcat <- factor(BankWages$education)
R> levels(edcat)[3:10] <- rep(c("14-15", "16-18", "19-21"),
+    c(2, 3, 3))
R> tab <- xtabs(~ edcat + job, data = BankWages)
R> prop.table(tab, 1)
```

```
       job
edcat    custodial    admin   manage
  8       0.245283 0.754717 0.000000
  12      0.068421 0.926316 0.005263
  14-15   0.008197 0.959016 0.032787
  16-18   0.000000 0.367089 0.632911
  19-21   0.000000 0.033333 0.966667
```

Visualize table in a spine plot via

```
R> plot(job ~ edcat, data = BankWages, off = 0)
```


Multinomial responses



Multinomial responses

Multinomial logit model is fitted via

```
R> library("nnet")
R> bank_mnl <- multinom(job ~ education + minority,
+    data = BankWages, subset = gender == "male", trace = FALSE)
```

Instead of `summary()` we just use

```
R> coeftest(bank_mnl)
```

```
z test of coefficients:

                     Estimate Std. Error z value Pr(>|z|)
admin:(Intercept)      -4.761      1.173   -4.06  4.9e-05
admin:education         0.553      0.099    5.59  2.3e-08
admin:minorityyes      -0.427      0.503   -0.85   0.3957
manage:(Intercept)    -30.775      4.479   -6.87  6.4e-12
manage:education        2.187      0.295    7.42  1.2e-13
manage:minorityyes     -2.536      0.934   -2.71   0.0066
```

Proportions of `"admin"` and `"manage"` categories (as compared with `"custodial"`) increase with `education` and decrease for `minority`. Both effects stronger for the `"manage"` category.

Ordinal responses

- Dependent variable `job` in multinomial example can be considered an ordered response: `"custodial"` $<$ `"admin"` $<$ `"manage"`.
- Suggests to try ordered logit or probit regression – we use ordered logit.
- Ordered logit model just estimates different intercepts for different job categories but common set of regression coefficients.
- Ordered logit often called proportional odds logistic regression (POLR) in statistical literature.
- `polr()` from **MASS** fits POLR and also ordered probit (just set `method="probit"`).

Ordinal responses

```
R> library("MASS")
R> bank_polr <- polr(job ~ education + minority,
+    data = BankWages, subset = gender == "male", Hess = TRUE)
R> coeftest(bank_polr)
```

```
z test of coefficients:

                Estimate Std. Error z value Pr(>|z|)
education         0.8700     0.0931    9.35  < 2e-16
minorityyes      -1.0564     0.4120   -2.56     0.01
custodial|admin   7.9514     1.0769    7.38  1.5e-13
admin|manage     14.1721     1.4744    9.61  < 2e-16
```

Results similar to (unordered) multinomial case, but different `education` and `minority` effects for different job categories are lost.

Appears to deteriorate the model fit:

```
R> AIC(bank_mnl)
```

```
[1] 249.5
```

```
R> AIC(bank_polr)
```

```
[1] 268.6
```