Limited Dependent Variable Models

Jerome Dumortier

30 March 2023

Overview

Packages and Files

Required packages:

- AER
- censReg
- foreign
- MASS
- pscl
- stargazer
- survival
- survminer
- truncreg

Required files:

```
data("NMES1988",package="AER")
```

Topics Covered

Regression models in which the dependent variable is somehow limited:

- Truncated data: Values above and/or below particular points are not reported
- Censored data: Values above and/or below particular points are reported at those points
- Count data: Discrete, integer count value
- Survival/duration data: Time to a certain event

Truncation

Overview

Concept

- Value above and/or below a certain point are not part of the data

Examples

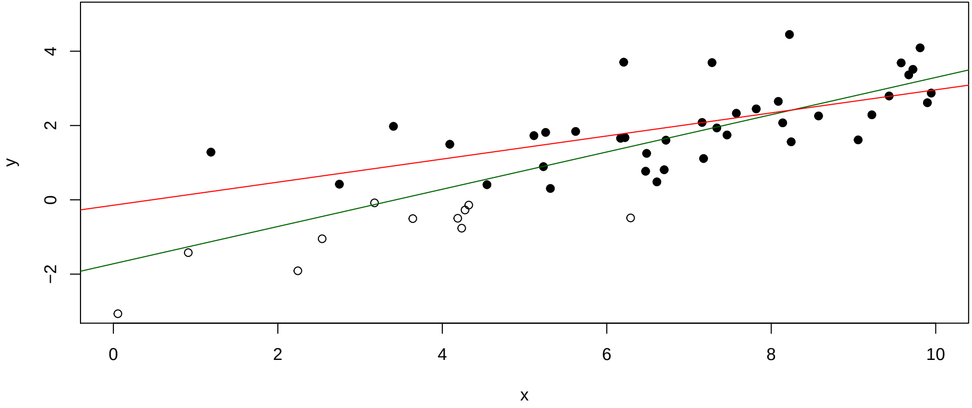
- Low income household studies
- On-site visitation data (unobserved non-visitors)
- Employment data on hours worked (excludes unemployed)

Simulated data

- "True" Coefficients: $\beta_0 = -2$ and $\beta_1 = 0.5$
- Values $y < 0$ are not reported in the data

Next slide: The green regression line is "correct" whereas the "red" is the line obtained from a regression model which ignores the truncation.

Graphical Illustration



Setup for truncation Data

```
truncation1     = truncation[c("y_real","x")]
truncation2     = subset(truncation,y_obs>0,select=c("y_obs","x"))
bhat_real       = lm(y_real~x,data=truncation1)
bhat_truncated = lm(y_obs~x,data=truncation2)
```

Required package to estimate a truncated model

- truncreg

Additional variable output *sigma*:

- Related to the truncated normal distribution

Complete vs. Truncated Data with Regular OLS

```
##
## ===============================================================
##                                Dependent variable:
##                     -------------------------------------------
##                           y_real                 y_obs
##                             (1)                   (2)
## ---------------------------------------------------------------
## x                        0.502***              0.311***
##                          (0.055)               (0.066)
## Constant                -1.721***              -0.146
##                          (0.368)               (0.483)
## ---------------------------------------------------------------
## Observations                50                    39
## R2                        0.634                 0.374
## Adjusted R2               0.626                 0.357
## Residual Std. Error   0.982 (df = 48)       0.854 (df = 37)
## F Statistic         83.110*** (df = 1; 48) 22.134*** (df = 1; 37)
## ===============================================================
## Note:                               *p<0.1; **p<0.05; ***p<0.01
```

Results: Correcting for Truncation

```
##
## Call:
## truncreg(formula = y_obs ~ x, data = truncation2, point = 0,
##     direction = "left")
##
## BFGS maximization method
## 22 iterations, 0h:0m:0s
## g'(-H)^-1g = 7.28E-09
##
##
##
## Coefficients :
##              Estimate Std. Error t-value  Pr(>|t|)
## (Intercept) -0.716370   0.668897 -1.0710    0.2842
## x            0.378806   0.086811  4.3636 1.279e-05 ***
## sigma        0.889160   0.119117  7.4646 8.349e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -46.315 on 3 Df
```

Achievement Scores: Data Load and Description

Loading the data using the package foreign

```
url         = "https://stats.idre.ucla.edu/stat/data/truncreg.dta"
achievement = read.dta(url)
```

Description of the data from UCLA Source:

"A study of students in a special GATE (gifted and talented education) program wishes to model achievement as a function of language skills and the type of program in which the student is currently enrolled. A major concern is that students are required to have a minimum achievement score of 40 to enter the special program. Thus, the sample is truncated at an achievement score of 40."

Achievement Scores: Regular OLS Estimation

```
##
## Call:
## lm(formula = achiv ~ langscore + prog, data = achievement)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9413  -5.7033  -0.8462   5.2205  21.3010
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.63965    3.70639   7.457 4.01e-12 ***
## langscore    0.46319    0.06792   6.820 1.45e-10 ***
## progacademic  2.97343    1.44889   2.052   0.0416 *
## progvocation -0.52118    1.72739  -0.302   0.7632
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.534 on 174 degrees of freedom
## Multiple R-squared:  0.3054, Adjusted R-squared:  0.2934
## F-statistic:  25.5 on 3 and 174 DF,  p-value: 1.01e-13
```

Achievement Scores: Truncated Model

```
##
## Call:
## truncreg(formula = achiv ~ langscore + prog, data = achievement,
##     point = 40, direction = "left")
##
## BFGS maximization method
## 57 iterations, 0h:0m:0s
## g'(-H)^-1g = 2.5E-05
##
##
##
## Coefficients :
##              Estimate Std. Error t-value  Pr(>|t|)
## (Intercept)  11.29942    6.77173  1.6686   0.09519 .
## langscore     0.71267    0.11446  6.2264 4.773e-10 ***
## progacademic  4.06267    2.05432  1.9776   0.04797 *
## progvocation -1.14422    2.66958 -0.4286   0.66821
## sigma         8.75368    0.66647 13.1343 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -591.31 on 5 Df
```

Censoring

Overview

Concept

- Value above and/or below a certain point are not part of the data

Examples

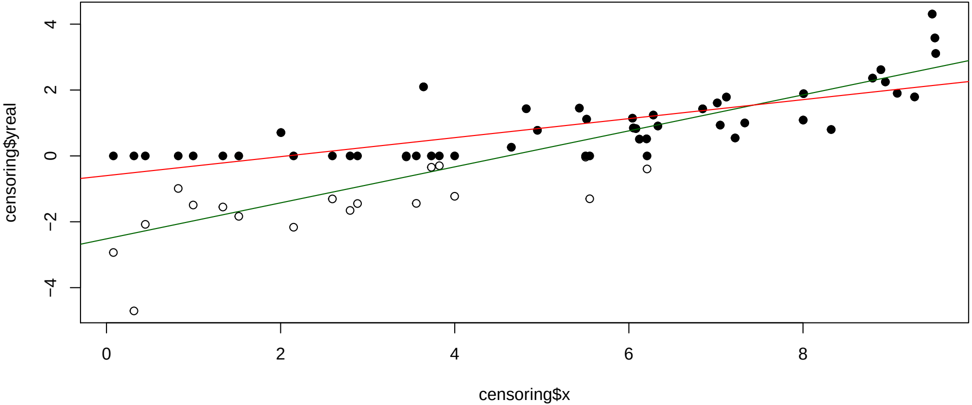
- Capacity constrained data, e.g., class enrollments or ticket sales
- Hours worked (or leisure demand), which is essentially capacity constrained
- Commodity purchases (non-negative)

Simulated data

- "True" Coefficients: $\beta_0 = -2$ and $\beta_1 = 0.5$
- Values $y < 0$ are reported at 0

R package censReg to reduce bias

Graphical Illustration



Complete vs. Censored Data with Regular OLS

```
##
## ============================================================
##                                   Dependent variable:
##                              ----------------------------
##                                  yreal           y
##                                   (1)           (2)
## ------------------------------------------------------------
## x                              0.546***      0.288***
##                                (0.047)       (0.035)
## Constant                      -2.518***     -0.599***
##                                (0.282)       (0.212)
## ------------------------------------------------------------
## Observations                      50            50
## R2                              0.737         0.579
## Adjusted R2                     0.732         0.570
## Residual Std. Error (df = 48)   0.907         0.685
## F Statistic (df = 1; 48)      134.839***    65.991***
## ============================================================
## Note:                         *p<0.1; **p<0.05; ***p<0.01
```

Estimation of a Censored Model

```
##
## Call:
## censReg(formula = y ~ x, data = censoring)
##
## Observations:
##          Total  Left-censored     Uncensored Right-censored
##             50             19             31              0
##
## Coefficients:
##             Estimate Std. error t value  Pr(> t)
## (Intercept) -2.10846    0.44764  -4.710 2.48e-06 ***
## x            0.48898    0.06582   7.429 1.09e-13 ***
## logSigma    -0.18013    0.12993  -1.386    0.166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Newton-Raphson maximisation, 6 iterations
## Return code 1: gradient close to zero (gradtol)
## Log-likelihood: -46.05041 on 3 Df
```

Count Models

Overview

Dependent variable

- Discrete, integer count data

Examples

- What are the number of arrests for a person?
- What determines the number of credit cards a person owns?

Three count data models

1. Poisson regression
2. Quasi-Poisson Regression Model
3. Negative Binomial Regression Model

Choice criteria: Presence or absence of overdispersion

- Overdispersion Variance of the dependent variable is larger than its mean.
- Poisson model is not suitable for overdispersion

Packages

The main package used is pscl. There is also an additional resource with more theoretical details on the topic: Regression Models for Count Data in R. A more up-to-date version of the document may be found with the pscl package documentation.

Poisson Regression Model

Recall Poisson distribution:

$$Pr(Y = k) = \frac{e^{-\lambda} \cdot \lambda^k}{k!}$$

Equidispersion as key characteristics:

- Mean and variance equal to λ, i.e., $E(Y) = \lambda$ and $Var(Y) = \lambda$
- Poisson regression: $\lambda = exp(\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_k \cdot x_k)$.

NHTS Example: Number of Vehicles (hhpub)

Data source

- 2017 National Household Travel Survey
- Survey quantifying trip and travel habits across the United States
- Example use: Quantifying intra-day electricity demand from electric vehicles

Outcome of interest

- Number of vehicles based on household income, home ownership, and urban/rural household location

Data preparation

- Elimination of missing and unknown data value
- Conversion of income to 1,000 dollars

Data Preparation

```
hhpubdata          = subset(hhpub,HHFAMINC %in% c(1:11) &
                            HOMEOWN %in% c(1,2) &
                            URBRUR %in% c(1,2) &
                            HHVEHCNT %in% c(0:12))
HHFAMINC           = c(1:11)
INCOME             = c(10,12.5,20,30,42.5,57.5,82.5,112.5,137.5,
                       175,200)
INCOME             = data.frame(HHFAMINC,INCOME)
hhpubdata          = merge(hhpubdata,INCOME)
hhpubdata$RURAL    = hhpubdata$URBRUR-1
hhpubdata$RENT     = hhpubdata$HOMEOWN-1
```

Poisson Model Execution

Preliminary step: Calculation of mean and variance of dependent variable

```
mean(hhpubdata$HHVEHCNT)
```

```
## [1] 1.981142
```

```
var(hhpubdata$HHVEHCNT)
```

```
## [1] 1.386027
```

Similar values and thus, Poisson regression model as an appropriate first step.

```
bhat_pois = glm(HHVEHCNT~INCOME+RENT+RURAL,
                data=hhpubdata,family=poisson)
```

Results

```
##
## Call:
## glm(formula = HHVEHCNT ~ INCOME + RENT + RURAL, family = poisson,
##     data = hhpubdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6889  -0.5568  -0.1558   0.3590   5.5063
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.654e-01  4.292e-03  108.43   <2e-16 ***
## INCOME       2.986e-03  3.601e-05   82.93   <2e-16 ***
## RENT        -3.733e-01  5.797e-03  -64.39   <2e-16 ***
## RURAL        2.224e-01  4.616e-03   48.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 86505  on 124400  degrees of freedom
## Residual deviance: 68533  on 124397  degrees of freedom
## AIC: 370161
##
## Number of Fisher Scoring iterations: 5
```

Interpretation

Sign of coefficients as an indication of the direction of influence on the outcome variable, i.e., the number of cars.

- Association of higher income and rural living with a higher number of car
- Association of renting with lower number of vehicles.
- Possible correlation between income and renting

General coefficient interpretation using $\exp(\beta)$, i.e., every unit increase in X has a multiplicative effect of $\exp(\beta)$ on the mean of Y, i.e., λ:

- $\beta = 0 \Rightarrow \exp(\beta) = 1$: Y and X are not related.
- $\beta > 0 \Rightarrow \exp(\beta) > 1$: Expected count $E(y)$ is $\exp(\beta)$ times larger than when $X = 0$
- $\beta < 0 \Rightarrow \exp(\beta) < 1$: Expected count $E(y)$ is $\exp(\beta)$ times smaller than when $X = 0$

Testing for Overdispersion I

Function `overdispersion()` from the package AER:

- Tests the null hypothesis of equidispersion (i.e., assuming no overdispersion)

Executed after the main regression using `glm(...,family=poisson)`

Testing for Overdispersion II

```
dispersiontest(bhat_pois)
```

```
##
##  Overdispersion test
##
## data:  bhat_pois
## z = -115.75, p-value = 1
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##  0.5670593
```

Given the p-value, the null hypothesis cannot be rejected. If the data suggests overdispersion, two alternative regression models can be used: (1) Quasi-Poisson and (2) Negative Binomial.

Quasi-Poisson Regression Model

Dataset `blm` from article Black Lives Matter: Evidence that Police-Caused Deaths Predict Protest Activity.

- Dependent variable: Total number of protests in a city
- Note that the paper includes a significant number of supplementary materials which allows for the replication of the results and much more.

First step: Calculation of mean and variance of the variable *totalprotests*:

```
mean(blm$totprotests)
```

```
## [1] 0.4959529
```

```
var(blm$totprotests)
```

```
## [1] 6.35326
```

Presence of Overdispersion

Likely overdispersion due to variance being significantly higher than mean. In a first step, a regular Poisson model is estimated.

```
eq1         = "totprotests~log(pop)+log(popdensity)+percentblack+
               blackpovertyrate+I(blackpovertyrate^2)+
               percentbachelor+collegeenrollpc+demshare"
eq2         = paste(eq1,"+deathsblackpc",sep="")
eq3         = paste(eq1,"+deathspc",sep="")
bhat1       = glm(eq1,data=blm,family=poisson)
bhat2       = glm(eq1,data=blm,family=quasipoisson)
```

Estimation Results

```
##
## ==========================================================
##                                Dependent variable:
##                      -------------------------------------
##                                    totprotests
##                           Poisson       glm: quasipoisson
##                                             link = log
##                             (1)                 (2)
## ----------------------------------------------------------
## log(pop)              1.129*** (0.040)   1.129*** (0.062)
## log(popdensity)      -0.183** (0.087)    -0.183 (0.135)
## percentblack          0.017*** (0.003)   0.017*** (0.005)
## blackpovertyrate      0.146*** (0.026)   0.146*** (0.041)
## I(blackpovertyrate2) -0.002*** (0.0004)  -0.002** (0.001)
## percentbachelor       0.039*** (0.004)   0.039*** (0.006)
## collegeenrollpc       0.009*** (0.002)   0.009** (0.004)
## demshare              0.043*** (0.005)   0.043*** (0.008)
## Constant             -20.009*** (0.633) -20.009*** (0.984)
## ----------------------------------------------------------
## Observations               1,226              1,226
## Log Likelihood           -612.473
## Akaike Inf. Crit.        1,242.946
## ==========================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
```

Note: Switch of statistical significance for population density

Testing for Overdispersion

```
##
##  Overdispersion test
##
## data:  bhat1
## z = 1.4052, p-value = 0.07998
## alternative hypothesis: true dispersion is greater than 1
## sample estimates:
## dispersion
##   2.212733
```

Null hypothesis rejected at 10% but not 5% significance level. The Quasi-Poisson Regression Model handles overdispersion by adjusting standard errors but leaving the coefficient estimates the same.

Negative Binomial Regression Model

The Negative Binomial Regression Model can be used in the presence of count data and overdispersion. Below, the results from the article Black Lives Matter: Evidence that Police-Caused Deaths Predict Protest Activity are recreated using the negative binomial models presented in the paper.

Three models:

1. Resource mobilization and opportunity structure
2. Adding black death
3. Adding all police-caused deaths instead (victims of any race)

BLM Models

```
bhat3 = glm.nb(eq1,data=blm,link=log)
bhat4 = glm.nb(eq2,data=blm,link=log)
bhat5 = glm.nb(eq3,data=blm,link=log)
```

BLM Model Results

```
##
## ===============================================================================
##                                            Dependent variable:
##                           -----------------------------------------------------
##                                               totprotests
##                                  (1)                (2)                (3)
## -------------------------------------------------------------------------------
## log(pop)                  1.292*** (0.072)   1.281*** (0.070)   1.277*** (0.071)
## log(popdensity)           -0.313** (0.133)   -0.305** (0.131)   -0.312** (0.132)
## percentblack              0.022*** (0.005)   0.018*** (0.005)   0.022*** (0.005)
## blackpovertyrate          0.132*** (0.031)   0.128*** (0.031)   0.129*** (0.031)
## I(blackpovertyrate2)    -0.001*** (0.0005) -0.001*** (0.0005) -0.001*** (0.0005)
## percentbachelor           0.045*** (0.005)   0.044*** (0.005)   0.045*** (0.005)
## collegeenrollpc           0.011** (0.004)    0.010** (0.004)    0.010** (0.004)
## demshare                  0.041*** (0.007)   0.041*** (0.007)   0.041*** (0.007)
## deathsblackpc                                2.825*** (0.931)
## deathspc                                                          0.956 (0.633)
## Constant                -20.905*** (1.117) -20.734*** (1.101) -20.801*** (1.108)
## -------------------------------------------------------------------------------
## Observations                  1,226              1,226              1,226
## Log Likelihood               -551.093           -546.677           -549.919
## theta                     1.559*** (0.351)   1.686*** (0.404)   1.622*** (0.374)
## Akaike Inf. Crit.            1,120.187          1,113.353          1,119.839
## ===============================================================================
## Note:                                               *p<0.1; **p<0.05; ***p<0.01
```

Hurdle and Zero-Inflation Models

Overview

Problem:

- Presence of many observations at 0 in count data
- Issues using Poisson or a Negative-Binomial Regression Model.

Application of hurdle and zero-inflated models:

- Data `NMES1988` from the package AER
- Data BLM protests

`NMES1988` Data:

- 4406 observations of people on Medicare who are 66 years or older.
- Outcome of interest: Number of doctor *visits*
- Independent variables: *hospital* (number of hospital visits), *health* (self-indicated health status), *chronic* (number of chronic conditions), *gender*, *school*, and *insurance*.

Estimation

```
eq          = visits~hospital+health+chronic+gender+
              school+insurance
bhat_pois   = glm(eq,data=NMES1988,family=poisson)
bhat_nb     = glm(eq,data=NMES1988)
bhat_hurdle = hurdle(eq,data=NMES1988,dist="negbin")
bhat_zi     = zeroinfl(eq,data=NMES1988)
```

Results

```
##
## ==============================================================
##                               Dependent variable:
##                 ----------------------------------------------
##                                    visits
##                  Poisson    normal    hurdle   zero-inflated
##                                                  count data
##                    (1)        (2)       (3)         (4)
## --------------------------------------------------------------
## hospital        0.165***  1.620***  0.212***    0.159***
##                 (0.006)   (0.133)   (0.021)     (0.006)
## healthpoor      0.248***  1.845***  0.316***    0.253***
##                 (0.018)   (0.312)   (0.048)     (0.018)
## healthexcellent -0.362*** -1.331*** -0.332***   -0.304***
##                 (0.030)   (0.363)   (0.066)     (0.031)
## chronic         0.147***  0.944***  0.126***    0.102***
##                 (0.005)   (0.077)   (0.012)     (0.005)
## gendermale      -0.112*** -0.632*** -0.068**    -0.062***
##                 (0.013)   (0.195)   (0.032)     (0.013)
## school          0.026***  0.143***  0.021***    0.019***
##                 (0.002)   (0.027)   (0.005)     (0.002)
## insuranceyes    0.202***  1.104***   0.100**    0.081***
##                 (0.017)   (0.244)   (0.043)     (0.017)
## Constant        1.029***  1.632***  1.198***    1.406***
##                 (0.024)   (0.335)   (0.059)     (0.024)
## --------------------------------------------------------------
## Observations      4,406     4,406     4,406       4,406
## ==============================================================
## Note:                            *p<0.1; **p<0.05; ***p<0.01
```

Survival Models

Overview

Length of time until a certain event occurs and variables influencing time passed (also known as time-to-event data analysis). Examples:

- Time to failure of mechanical device
- Time to death after diagnosis with a certain disease
- Time to re-arrest after release from prison
- Time to defaulting on loan or mortgage

Data used for this topic

- `rossi`

Theoretical Aspects

T as a random variable representing survival time with the cumulative distribution function written as:

$$F(t) = Pr(T \leq t)$$

where t is a realization of T. Survival function as the complement probability (at least t) :

$$S(t) = 1 - F(t) = Pr(T \geq t)$$

Hazard function or hazard rate $h(t)$ as risk of failure at time t.

Example Data `rossi`

Experimental recidivism study on 432 male prisoners over a period of one year after release from prison (Rossi et al., 1980):

- *week*: Week of first arrest after release
- *arrest*: Event indicator equal to 1 for rearrest during study period
- *fin*: Receipt of financial aid after release from prison (randomly assigned factor by the researchers)
- *age*: Age at the time of release
- *race*: Black and other
- *wexp*: Full-time work experience prior to incarceration
- *mar*: Married at the time of release
- *paro*: Released on parole
- *prio*: Number of prior convictions.
- *educ*: Education coded as 2 (grade 6 or less), 3 (grades 6-9), 4 (grades 10-11), 5 (grade 12), or 6 (some post-secondary).

Analysis

Number of prisoners rearrested during study period:

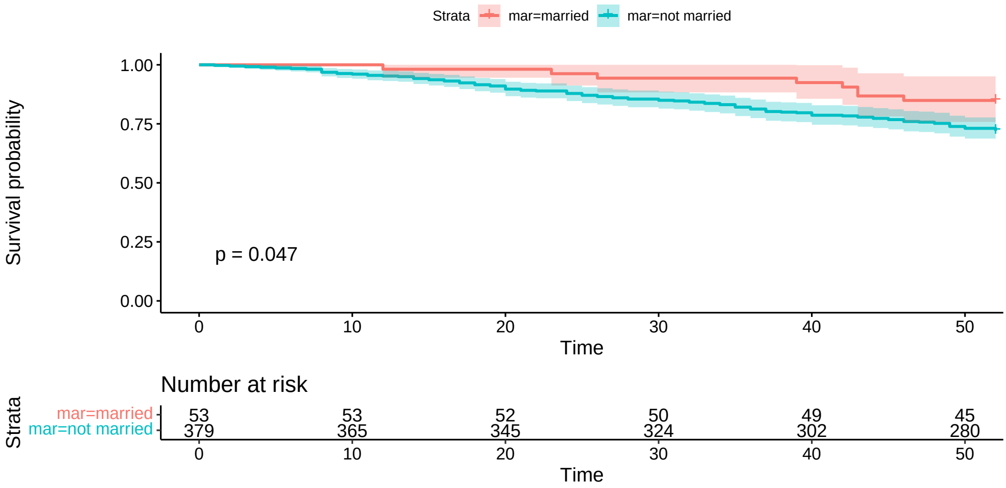
```
sum(rossi$arrest)
```

```
## [1] 114
```

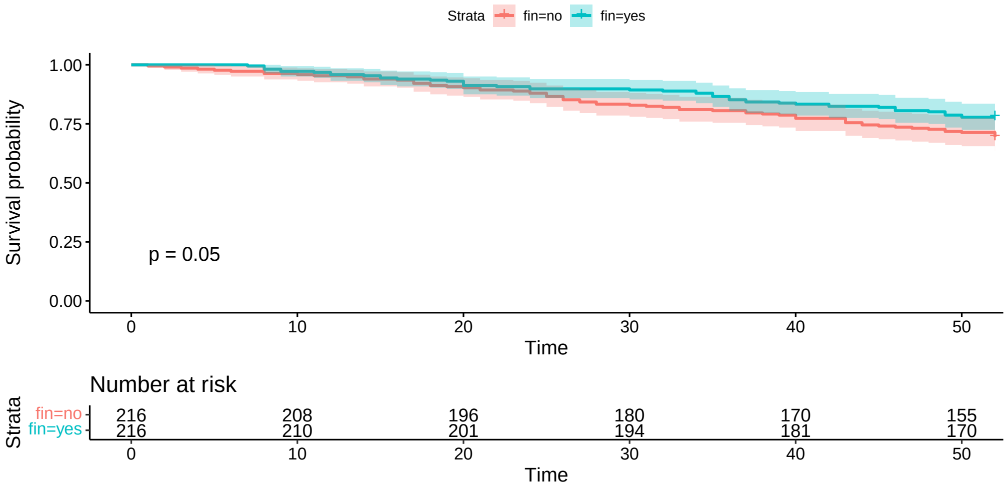
Surival object in R created by function `Surv()`:

```
bhatmar = survfit(Surv(week,arrest)~mar,data=rossi)
bhatfin = survfit(Surv(week,arrest)~fin,data=rossi)
ggsurvplot(bhatmar,pval=TRUE,risk.table=TRUE)
ggsurvplot(bhatfin,pval=TRUE,risk.table=TRUE)
```

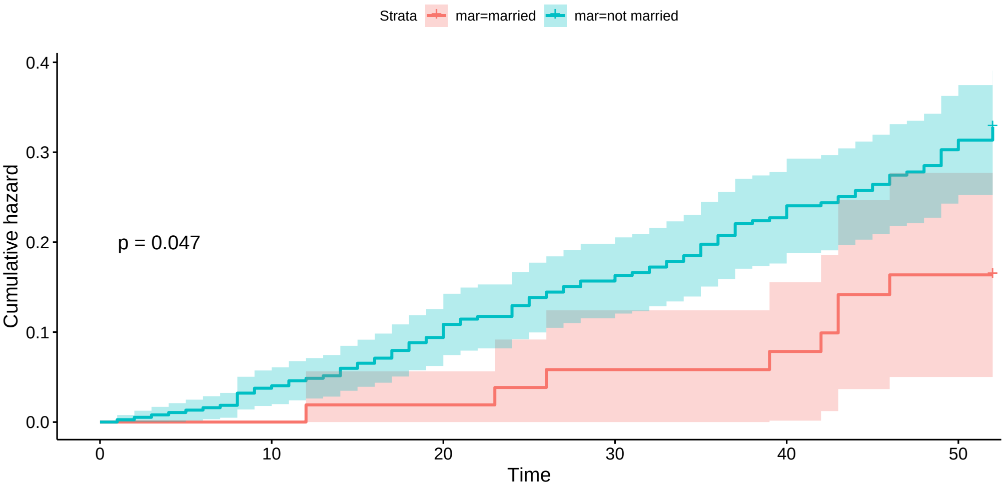
Survival Curve: Marriage



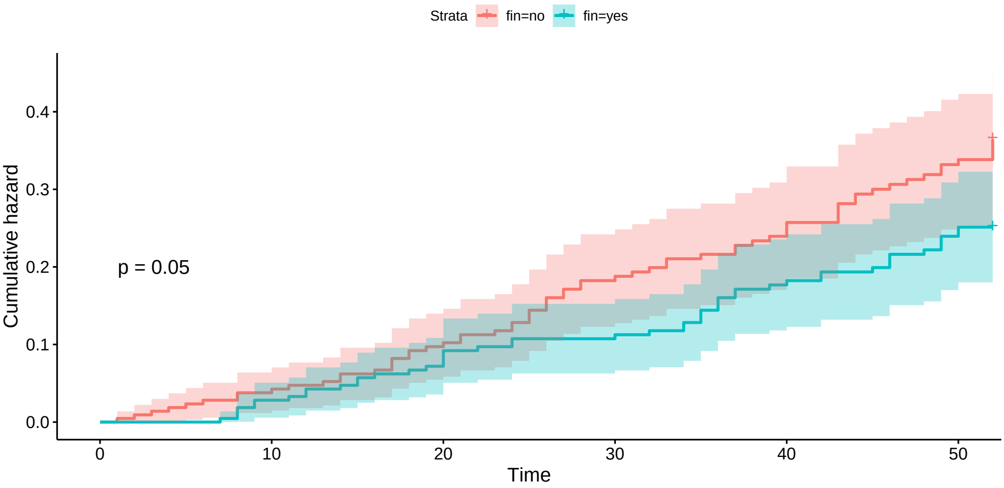
Survival Curve: Financial Aid



Cumulative Hazard Function: Marriage



Cumulative Hazard Function: Financial Aid



Cox Regression in R: Setup

```
bhat1 = coxph(Surv(week,arrest)~mar,data=rossi)
bhat2 = coxph(Surv(week,arrest)~mar+fin,data=rossi)
bhat3 = coxph(Surv(week,arrest)~fin+age+race+wexp+mar+paro+prio,
              data=rossi)
```

Statistically insignificant variables excluded from regression output on next slide due to space constraints: *paroyes*, *raceother*, and *wexpyes*

- In general, all variables must be reported!

Cox Regression in R: Results

```
##
## ==========================================================================
##                                    Dependent variable:
##                      -----------------------------------------------------
##                                           week
##                            (1)              (2)               (3)
## --------------------------------------------------------------------------
## marnot married           0.712*          0.738**            0.434
##                         (0.367)          (0.367)           (0.382)
## prio                                                      0.091***
##                                                           (0.029)
## finyes                                  -0.387**          -0.379**
##                                          (0.190)           (0.191)
## age                                                       -0.057***
##                                                           (0.022)
## --------------------------------------------------------------------------
## Observations               432              432               432
## R2                        0.011            0.020             0.074
## Max. Possible R2          0.956            0.956             0.956
## Log Likelihood          -673.060         -670.955          -658.748
## Wald Test            3.770* (df = 1)  7.930** (df = 2) 32.110*** (df = 7)
## LR Test              4.642** (df = 1) 8.852** (df = 2) 33.266*** (df = 7)
## Score (Logrank) Test 3.935** (df = 1) 8.139** (df = 2) 33.529*** (df = 7)
## ==========================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```