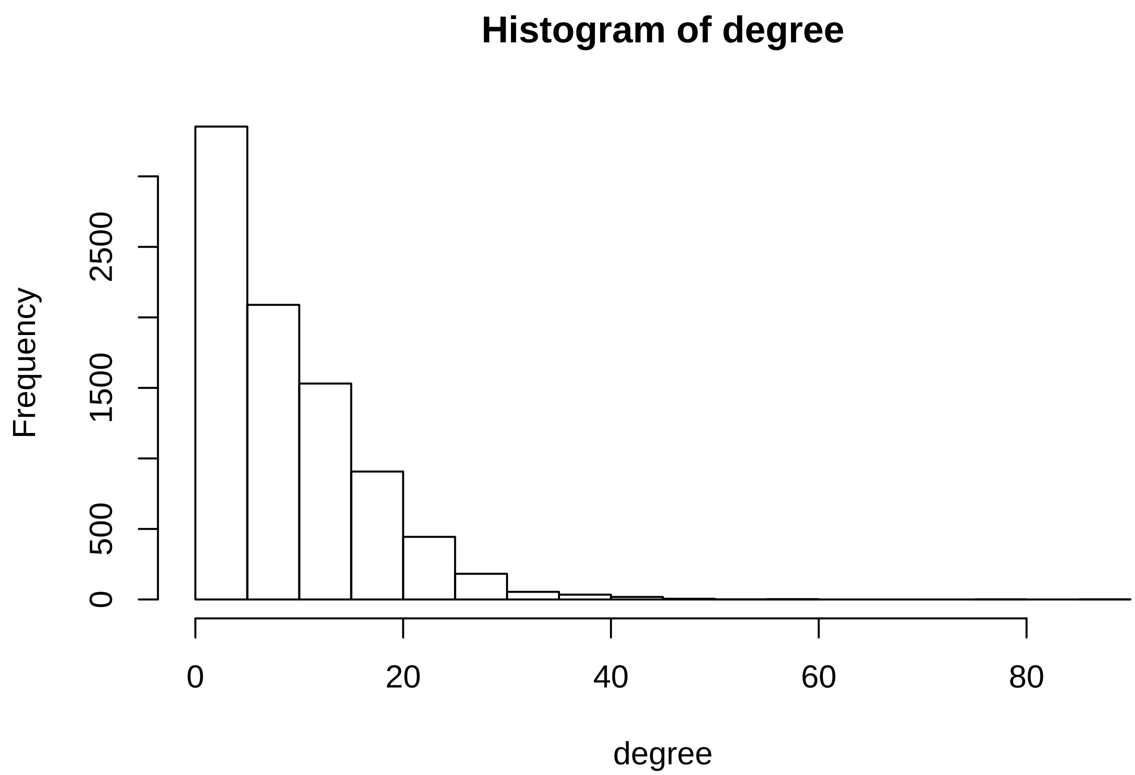
Village Connections: Will You Borrow From Your Neighbor?

(Student: Vinh Luong - 442069)

1. TRANSFORMATION OF DEGREES TO LOG SCALE

First of all we observe that the *degree* variable seems to display an exponential distribution:

```
hist(degree)
```



It is hence reasonable to transform this variable to the log scale and interprete its effects multiplicatively instead of additively:

```
log_degree <- log(degree + 1)
```

2. REGRESSION OF LOG(DEGREE) ON DEPENDENT VARIABLES

We now run a LASSO regression of the *log(degree)* variable on the dependent variables:

```r
m <- model.matrix(loan ~ ., data=hh)
m <- m[, -1] #exclude intercept

# LASSO REGRESSION
log_degree_regression <- gamlr(m, log_degree,
  family="gaussian", standardize=TRUE)

fitted_values <- as.vector(predict(log_degree_regression, m))
```

The correlation between the fitted values and the *log_degree* variable is low, only 0.2861384, suggesting that there is a significant potential treatment effect contained in *log_degree*.

3. REGRESSION TO DETECT TREATMENT EFFECT

We now run a LASSO regression of the *loan* variable on the dependent variables including the *log(degree)* variable and the fitted values from the regression in Section 2. We do not penalize the fitted values in order to fully capture the treatment effect:

```r
m1 <- cbind(log_degree, fitted_values, m)
loan_regression_1 <- gamlr(m1, hh$loan,
                           free = 2,
                           family = 'binomial', standardized=TRUE)
bAICc1 <- coef(loan_regression_1)
bAICc1
```

```
## 55 x 1 sparse Matrix of class "dgCMatrix"
##                          seg78
## intercept           -0.87072563
## log_degree           0.16088510
## fitted_values       -0.39360794
## village2             .
## village3            -0.32129420
## village4            -0.81497449
## village12            0.63089802
## village19           -0.54424199
## village20            1.00810593
## village21            0.30578905
## village23            0.05586155
## village24            0.06204849
## village25            0.13166061
## village28           -0.73655384
## village29            0.41528818
## village31            0.77579559
## village32            0.13405341
## village33           -0.17658261
## village36           -1.03358161
## village39           -0.51224647
## village42            0.79301843
## village43            0.14851000
## village45            0.31764378
## village46           -1.05343548
## village47            0.19904452
```

```
## village50             -0.93031601
## village51             -0.30264027
## village52             -0.40753477
## village55             -0.61630059
## village57              .
## village59             -1.02832253
## village62             -0.35113611
## village65              .
## village67              0.82386680
## village68             -0.26219169
## village70              .
## village71             -0.51992900
## village72             -0.15406979
## village73              0.12502330
## village75              0.15024314
## religionhindu         -0.56243219
## religionislam          .
## roofrcc               -0.17588701
## roofsheet              0.11467755
## roofstone              .
## roofthatch             0.24547637
## rooftile               0.16783843
## rooms                 -0.03477761
## beds                  -0.06997975
## electricity            0.23805486
## ownershipLEASED        .
## ownershipOWNED        -0.06237562
## ownershipRENTED       -0.07526373
## ownershipSHARE_OWNED -0.16796877
## leader                 0.54827717
```

The above results suggest that *log(degree)* has a positive treatment effect (coefficient 0.1608851) on the odds of a household getting a loan.

4. NAIVE LASSO

Let's compare the above results with those from a naive LASSO regression of the *loan* variable on the dependent variables and the *log(degree)*, but without including the fitted values:

```
m2 <- cbind(log_degree, m)
loan_regression_2 <- gamlr(m2, hh$loan,
                         family = 'binomial', standardized=TRUE)
bAICc2 <- coef(loan_regression_2)
bAICc2
```

```
## 54 x 1 sparse Matrix of class "dgCMatrix"
##                            seg94
## intercept             -1.37540986
## log_degree             0.16609976
## village2               0.03183144
## village3              -0.36612100
## village4              -0.87591382
## village12              0.63084954
```

```
## village19            -0.62715638
## village20             1.11558562
## village21             0.25264284
## village23             0.05886219
## village24             0.02116727
## village25             0.11113182
## village28            -0.77520187
## village29             0.40626119
## village31             0.74391430
## village32             0.10678119
## village33            -0.19544574
## village36            -1.12545048
## village39            -0.53802738
## village42             0.82040892
## village43             0.13752159
## village45             0.32298443
## village46            -1.14370335
## village47             0.16701634
## village50            -1.02283273
## village51            -0.44567042
## village52            -0.55975851
## village55            -0.61742793
## village57             .
## village59            -1.07036684
## village62            -0.42934666
## village65            -0.13865421
## village67             0.71555361
## village68            -0.37284679
## village70            -0.17967311
## village71            -0.63386808
## village72            -0.29671966
## village73             .
## village75             0.06993543
## religionhindu        -0.63142544
## religionislam         .
## roofrcc              -0.16977239
## roofsheet             0.16585936
## roofstone             .
## roofthatch            0.30564923
## rooftile              0.20227447
## rooms                -0.06352550
## beds                 -0.07781337
## electricity           0.16014441
## ownershipLEASED       0.06392337
## ownershipOWNED       -0.12278426
## ownershipRENTED      -0.05476866
## ownershipSHARE_OWNED -0.19681422
## leader                0.44045847
```

The coefficent 0.1660998 is very similar to what we got in Section 3. This is perhaps because the variable *log(degree)* is largely uncorrelated with the other dependent variables and hence the treatment effect is largely unaffected even if we do not include the fitted values.

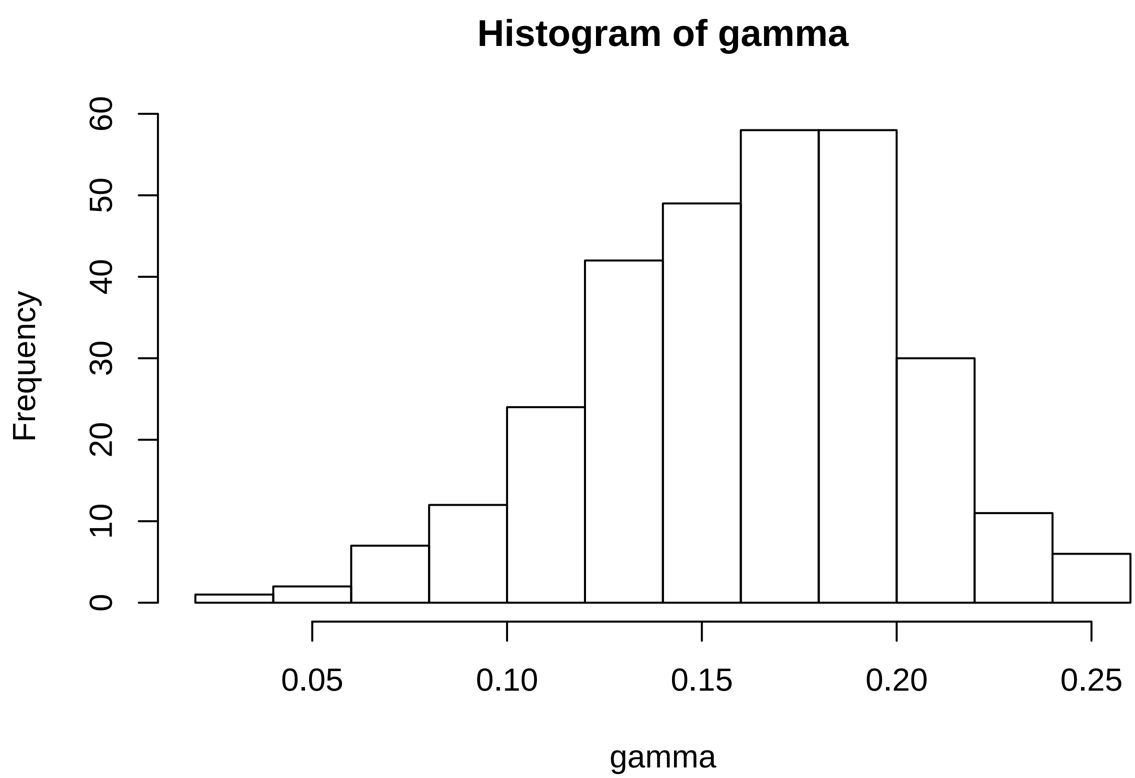
5. BOOTSTRAPPING

Let's now quantify the uncertainty in the treatment effect size by bootstrapping:

```
gamma <- c()
n <- nrow(m)
n
```

```
## [1] 8622
```

```
for(b in 1:300)
{
  ib <- sample(1 : n, n, replace=TRUE)
  m1 <- cbind(log_degree, fitted_values, m)[ib, ]
  loan_regression_1 <- gamlr(m1, hh$loan[ib],
                              free = 2,
                               family = 'binomial', standardized=TRUE)
  gamma <- c(gamma,coef(loan_regression_1)['log_degree', ])
}
hist(gamma)
```



The bootstrap suggests that the size of the treatment effect has a mean of about 0.1608426 and standard deviation of about 0.0409578.