HW3

Thomas Le, ID: 913091973 ; Armand Nasseri, ID: 912679383

May 30, 2019

We are taking one of our free late days.

A

NOTE: Written A is attached at the bottom of the pdf.

B

Cov(D, N) = E[DN] - (E[D] * E[N])

$E[DN] = \sum_{i=1}^{10} i(11-i)(1-0.15)^{i-1}(0.15) + \sum_{i=11}^{\infty} i(i-11)0.85^{i-1}0.15$

$E[D] = \sum_{i=1}^{10}(11-i)(1-0.15)^{i-1}(0.15) + \sum_{i=11}^{\infty}(i-11)0.85^{i-1}0.15$

$E[N] = \frac{1}{p}$

Then, $Cov(D,N) = \sum_{i=1}^{10} i(11-i)(1-0.15)^{i-1}(0.15) + \sum_{i=11}^{\infty} i(i-11)0.85^{i-1}0.15 - [\sum_{i=1}^{10}(11-i)(1-0.15)^{i-1}(0.15) + \sum_{i=11}^{\infty}(i-11)0.85^{i-1}0.15 * \frac{1}{p}]$

Plugging in p = 0.15 into a calculator =>

(17.14228235171777 + 26.031171240606657) - ((5.448954957930762 + 1.115621624597426) * (6.666667)) = -0.5903925

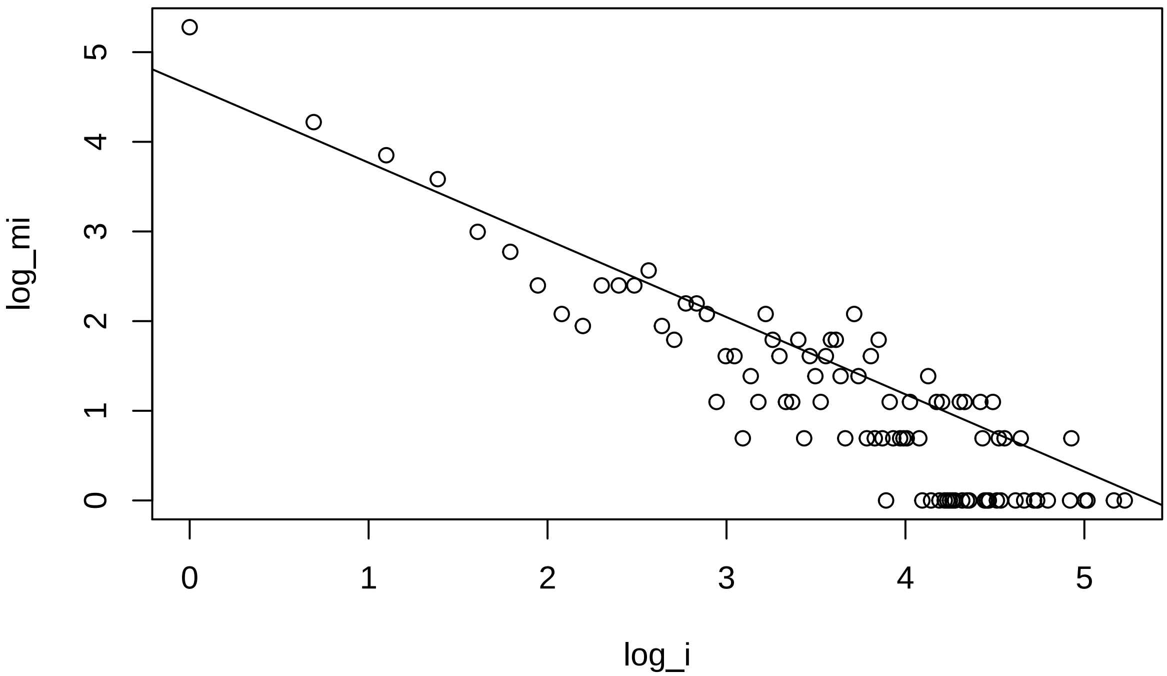
C

```r
library(plyr)

# get data, skip header row
original_data <- read.table("./dnc-corecipient/out.dnc-corecipient", skip=1)

# rename cols 1 and 2
colnames(original_data) <- c("id1", "id2", "nummsgs")

# select rows where id1 < id2 to remove duplicate data, keeping only cols 1 and 2
dnc <- original_data[original_data$id1 < original_data$id2, c(1,2)]

# create empty vector of 0s; get degrees by counting num occurences of each val
degrees <- rep(0, nrow(dnc))
for (i in 1:nrow(dnc)) {
  degrees[dnc[i, "id1"]] <- degrees[dnc[i, "id1"]] + 1
  degrees[dnc[i, "id2"]] <- degrees[dnc[i, "id2"]] + 1
}
```

```r
# get max degree, to know what i goes up to
max_degree <- max(degrees)

# mi is count of recipients having degree i
mi <- rep(0, max_degree)
for (i in 1:nrow(dnc)) {
  # increment its count
  mi[degrees[i]] <- mi[degrees[i]] + 1
}

# since calling log on 0 returns -Inf, we replace 0s with NA so plot will ignore the NA vals
for (i in 1:length(mi)) {
  if (mi[i] == 0)
    mi[i] <- NA
}

# i goes from 1 to max degree
i <- c(1:max_degree)

# apply log to i and mi
log_i <- log(i)
log_mi <- log(mi)

# apply linear model function and plot
lm(log_i ~ log_mi)
```

```
##
## Call:
## lm(formula = log_i ~ log_mi)
##
## Coefficients:
## (Intercept)       log_mi
##      4.6287      -0.8614
```

```r
plot(log_i, log_mi)
abline(lm(log_i ~ log_mi))
```




```r
summary(lm(log_i ~ log_mi))
```

```
##
## Call:
## lm(formula = log_i ~ log_mi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94058 -0.31176 -0.06983  0.32264  0.89563
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.62867    0.06505   71.15   <2e-16 ***
## log_mi      -0.86135    0.04189  -20.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.427 on 90 degrees of freedom
##   (94 observations deleted due to missingness)
## Multiple R-squared:  0.8245, Adjusted R-squared:  0.8226
## F-statistic: 422.8 on 1 and 90 DF,  p-value: < 2.2e-16
```

An estimate for gamma was found by applying a linear model function onto the logarithmic values of m_i and i, where m_i denotes the count of recipients having degree i in the data, $i = 1, 2, 3 \ldots$.

Gamma is estimated from the linear model to be about -0.8616, the slope from the linear fit. Interpreting that in context of the data means that as the degree i grows larger (i.e. number of times a recipient is inolved in a message), then m_i (count of unique recipients being involved in i messages) grows smaller. In simpler terms, it is a small amount of people that are involved in the most messages, which matches the description of a power law.

Viewing the summary of the linear model, we see that the p-value of 2.2e-16 is less than 0.05. Thus, the model is statistically significant and that the data fits the power law distribution well enough.