Homework 3 - Problem A

Rajat Mahesh Gupta

2024-12-01

Download the DNC e-mail co-recipient dataset from canvas. Here we will explore the degree distribution of the resulting graph.

- The format seems to be
  recip1ID recip2ID nummsgs
- One might treat this as a weighted graph, but we will not do so. Ignore the third column.
- The data seems to be duplicated, with (a,b) and (b,a) both appearing in the data. This is odd, as the third column differs, lets take only the rows in which recip1ID < recip2ID. This indeed cuts the data in half.
- Explore how well a power law fits the data, as follows. Let mi denote the count of recipients having degree i in the data, i = 1,2,3,… The form of the pmf (check this!) implies that a plot of log(mi) against log(i) should look like a straight line. There will be points above and below the line, due to sampling variation, but the trend should look linear. Make this plot, and comment.
- NOTE: In this and all future work for the course, you must use R to generate your graphs. This can be either base R, ggplot2 (appendix in our book) or lattice.
- Assuming a power law, estimate γ. (We use the term estimate here because the data is only a sample from a population (real or conceptual). Use R's lm() function for this. We will study this function in detail later, but for instance the following would find the intercept and slope of a line fit through the points (1,1), (2,2), (3,4):

```
> x <- 1:3
> y <- c(1,2,4)
> lm(y ~ x)
```

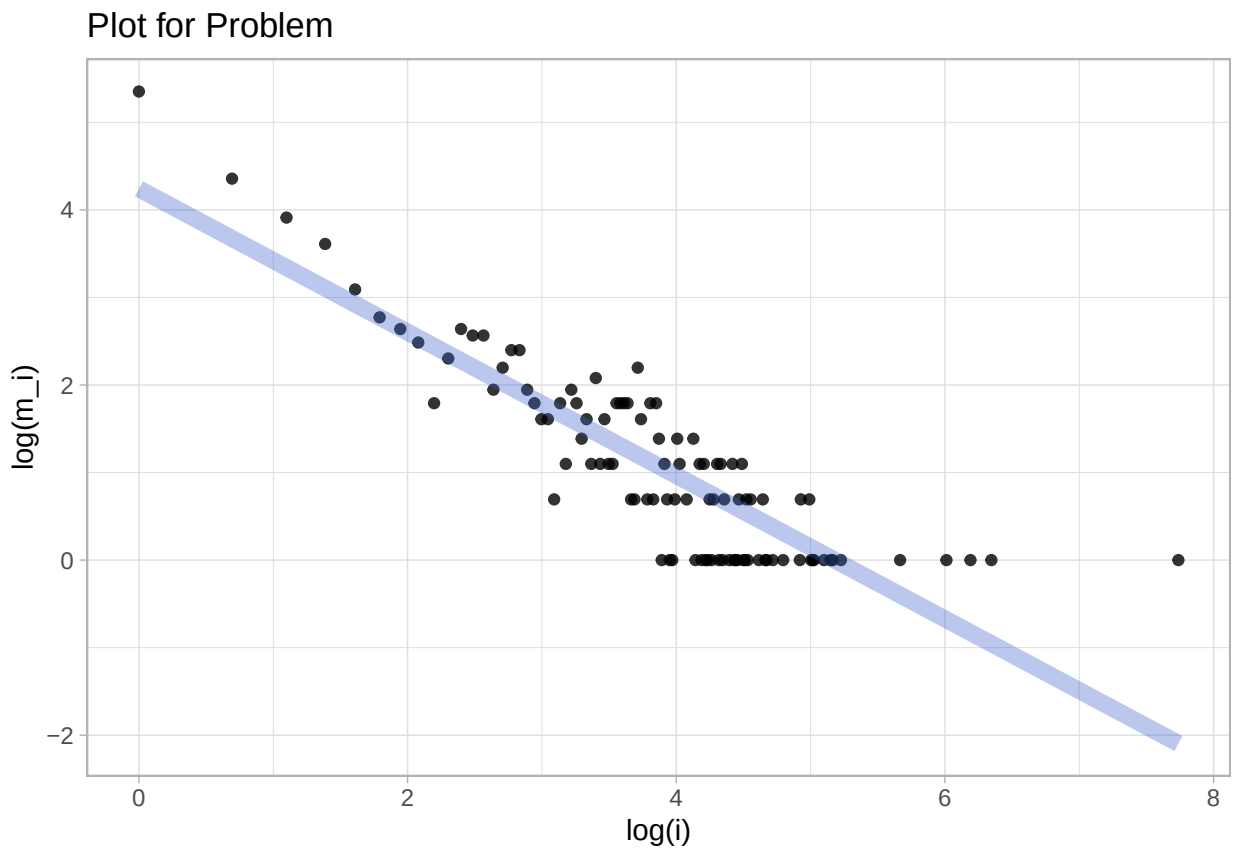
```r
library(ggplot2)

dat <- read.table("/Users/rajat-mahesh-gupta/Documents/csSchool/ECS-132/Week 9/HW 3/Problem A/dnc-corec

# keyword names used to rename the columns
names(dat) <- c("recip1ID", "recip2ID", "num_messages")

# Removes the duplicate entries where recip1ID < recip2ID
dat <- dat[dat["recip1ID"] < dat["recip2ID"], ]

# to find the degree of each recipient we call the table function on the column recip1ID
# table(as.matrix(dat)) would return something like { 1: 3, 2: 5, 3: 1 ... }
# calling table() again on table(as.matrix(dat)) would count the frequencies of the frequencies giving
```

```r
i_mi_dat <- table(table(as.matrix(dat)))

i = as.numeric(names(i_mi_dat))
mi <- as.vector(i_mi_dat)

data_to_plot <- data.frame(log_i = log(i), log_m_i = log(mi))
names(data_to_plot) <- c("log_i", "log_m_i")

# plotting
p <- ggplot(data_to_plot, aes(x=log_i, y=log_m_i)) + geom_point(color="black", alpha=0.8) + labs(title=
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
plot(p)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Save the plot
ggsave("problemA.png")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

- Re-do your earlier plot with the fitted line superimposed. Do not hard-code the intercept and slope from above; instead, call coef() on the object returned by lm()

```
# Print fit coefficients
linear_fit <- lm(log_m_i ~ log_i, data_to_plot)
print(coef(linear_fit))
```

```
## (Intercept)       log_i
##   4.2407828  -0.8187382
```

If the degree of a person follows the power law distribution, then we would expect a plot of log(mi) vs. log(i) to be a straight line. Figure 1 shows this plot for our dataset. The slope of this line would be γ. Thus, an estimate of γ for our data is log_i (from our linear fit coefficient). We can see the graph has a linear trend, but many points are not on the line. This is due to sampling variation - common in real world data.