MSD 2019 Final Project

A replication and extension of Systematic Inequality and Hierarchy in Faculty Hiring Networks by Aaron Clauset, Samuel Arbesman, Daniel B. Larremore, Science Advances 12 Feb 2015, Vol. 1, No. 1

George Austin (gia2105), Calvin Tong (cyt2113), Mia Fryer (mzf2106)

2019-05-11 13:06:24

Contents

Introduction

This report is a recreation and extension of the paper "Systematic inequality and hierarchy in faculty hiring networks" by Aaron Clauset, Samuel Arbesman and Daniel Larremore published in Science Advances in 2015. The paper explores the role of gender and institutional prestige in the faculty job market and finds that it is a deeply hierarchical and unequal system. Analyzing these effects is important as institutional hiring and faculty quality affects all aspects of university life both for the hired scholars as well as their undergraduate students. Furthermore, universities are often ideally seen as meritocracies with tests, grades, and journal publications allowing one's personal ability, as opposed to their networking ability, to shine. Debunking this myth will allow us to make the changes necessary to move closer to this ideal in the future.

The main contribution of the paper is the dataset, which has been painstakingly scraped and cleaned from a multitude of different sources. The data represents the placement of 18924 different faculty members at 461 academic institutions across the disciplines of Business, Computer Science and History. To explore the hierarchical structure of the departmental networks, the paper presents various figures and computations, which quantify and describe the inequality in different ways. For this report, we choose to recreate what we see as the most convincing of these results. Below we recreate, the visualization of the placement network for the top 10 schools in each department (Fig 1 top), the Lorentz curves for each department (Fig 2A), the network visualizations of the entire network for each department with the top 15% of schools highlighted (Fig 3A), and the probability distributions of the relative change in prestige rank for the top 15% of schools (Fig

3B) and all other institutions (Fig 3C). We also recreate the calculate the Gini Coefficient, which quantifies the social inequality in the network. For our extension, we introduce some predictive results to augment the primarily descriptive results from the paper. To do this, we attempt to predict the rank and prestige of the hiring party from different factors and interpret the coefficients to decipher overall trends from the data.

```r
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0       v purrr   0.3.0
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.2       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.3.0

## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(modelr)
library(ggplot2)
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following object is masked from 'package:modelr':
##
##     permute

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:purrr':
##
##     compose, simplify

## The following object is masked from 'package:tidyr':
##
##     crossing

## The following object is masked from 'package:tibble':
##
##     as_data_frame

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```r
library(reldist)
```

```
## reldist: Relative Distribution Methods
## Version 1.6-6 created on 2016-10-07.
## copyright (c) 2003, Mark S. Handcock, University of California-Los Angeles
##  For citation information, type citation("reldist").
##  Type help(package="reldist") to get started.
```

```r
library(here)
```

```
## here() starts at /Users/calvin/Documents/Columbia/msd2019-final-project-final-project-group-9
```

```r
library(modelr)
```

Data Loading

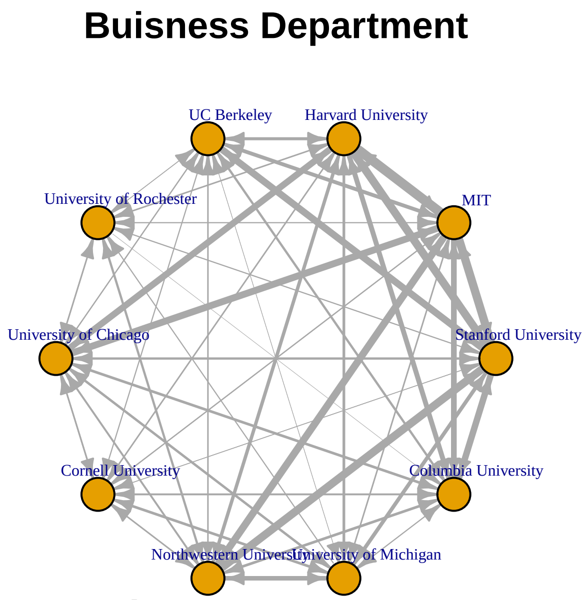
The data provided by the authors required no additional cleaning to reproduce the results. Here we load the data individually and store it in a list to allow for exploration and efficient iteration.

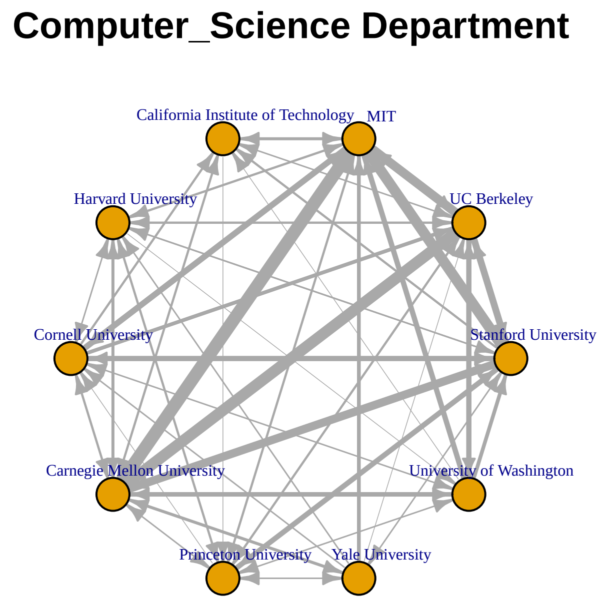
```r
fp_prefix <- "data/original/"

# Read by department data individually

business_edglist <- read.table(paste(fp_prefix, "Business_edgelist.txt", sep = ""),
  header = FALSE,
  col.names = c("u", "v", "rank", "gender")
)
business_vertexlist <- read.table(
  file = paste(fp_prefix, "Business_vertexlist.txt", sep = ""),
  sep = "	", header = FALSE,
  col.names = c("u", "pi", "USN2009", "NRC2010", "Region", "institution")
)

computer_science_edglist <- read.table(paste(fp_prefix, "ComputerScience_edgelist.txt", sep = ""),
  header = FALSE,
  col.names = c("u", "v", "rank", "gender")
)
computer_science_vertexlist <- read.table(
  file = paste(fp_prefix, "ComputerScience_vertexlist.txt", sep = ""),
  sep = "	", header = FALSE,
  col.names = c("u", "pi", "USN2009", "NRC2010", "Region", "institution")
)

history_edglist <- read.table(paste(fp_prefix, "History_edgelist.txt", sep = ""),
  header = FALSE,
  col.names = c("u", "v", "rank", "gender")
)
history_vertexlist <- read.table(
  file = paste(fp_prefix, "History_vertexlist.txt", sep = ""),
  sep = "	", header = FALSE,
  col.names = c("u", "pi", "USN2009", "NRC2010", "Region", "institution")
)

# Store data in list structure for iterations

data_list <- list(
  "Buisness" = list("edge" = business_edglist, "vert" = business_vertexlist),
  "Computer_Science" = list("edge" = computer_science_edglist, "vert" = computer_science_vertexlist),
  "History" = list("edge" = history_edglist, "vert" = history_vertexlist)
)
```

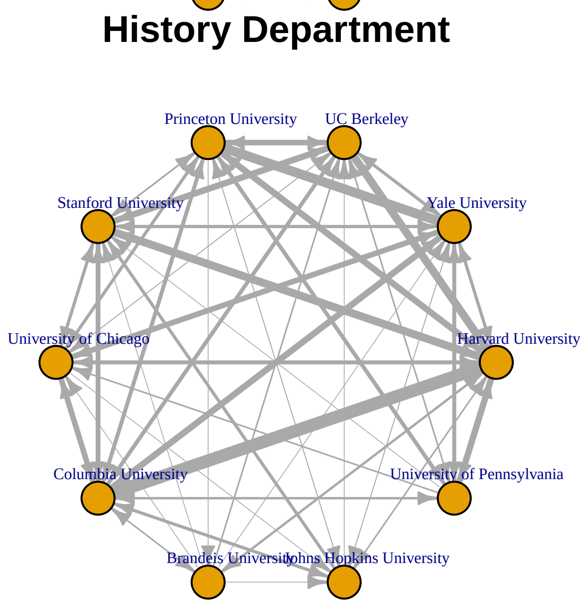
Produce Network Visualizations (Fig 1 Top)

We begin by reproducing the network visualizations for each department. The paper does not report the graphs for all the departments, but we have created them below to allow for better understanding of the network shapes. Since we are plotting all the departments, we have to normalize the edge sizes to make the plots comparable. The paper did not mention any normalization as they were only showing a single department. We decided to normalize by dividing through by the total number of edges.

```
for (dep in names(data_list)) {
  edgelist <- data_list[[dep]]$edge
  vertex <- data_list[[dep]]$vert

  # Making table that includes the weight of each edge
  weighted_edgelist <- edgelist %>%
    group_by(v, u) %>%
    summarize(count = n()) %>%
    ungroup() %>%
    left_join(vertex, by = c("v" = "u")) %>%
    select(v, u, count, institution)


  # Fitlering the weighted edgelist
  smaller <- weighted_edgelist %>%
    filter(u <= 10, v <= 10, u != v)

  # Plotting network of the top schools
  smaller_graph <- smaller %>%
    graph_from_data_frame(directed = TRUE)

  plot(smaller_graph,
    vertex_size = 0.5,
    edge.width = E(smaller_graph)$count / sum(E(smaller_graph)$count) * 100,
    edge.arrow.size = 0.5,
    layout = layout_in_circle(smaller_graph, order = V(smaller_graph)),
    vertex.label = unique(E(smaller_graph)$institution),
    vertex.label.cex = c(0.5),
    vertex.label.dist = 2,
    main = paste(dep, "Department", sep = " ")
  )
}
```





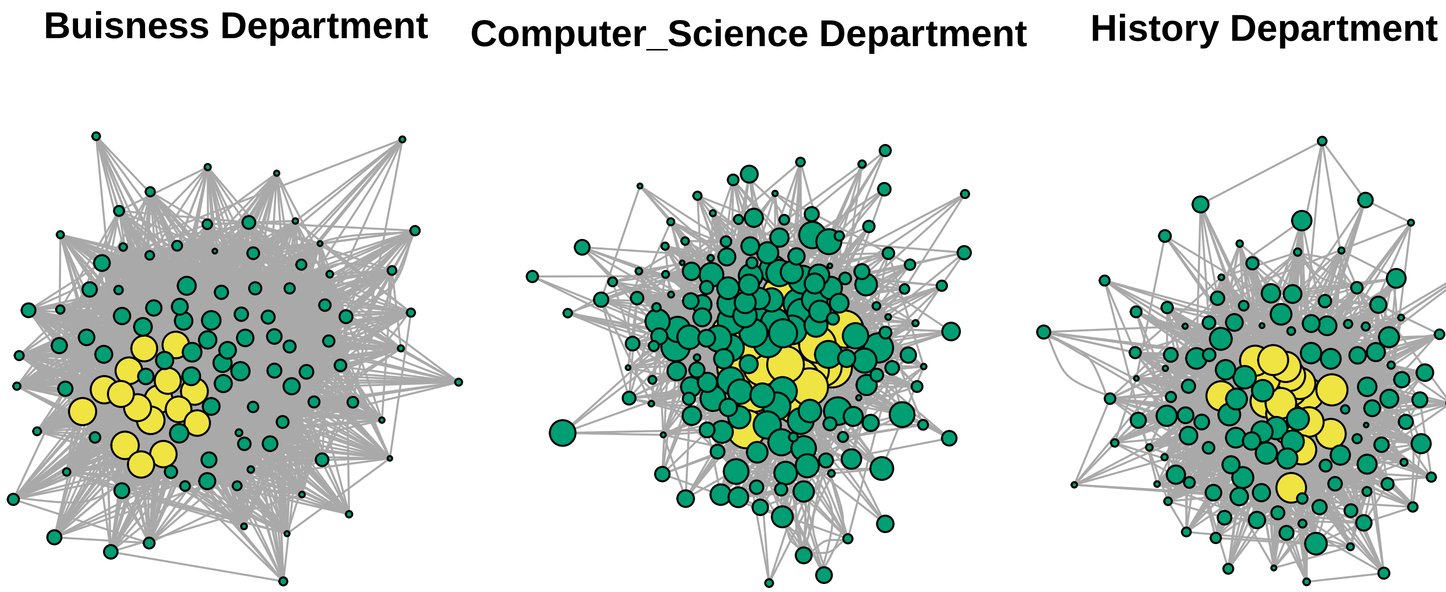


For comparison to the paper, we see that MIT's Computer Science department sends the most graduates to Carnegie Mellon. We see that for the history departments, there are strong ties between Harvard and Columbia and Princeton and Yale. We also see there is a strong ivy league network and dominance in the department, which we don't see in the Computer Science or Business networks. For Business schools, we see a much more balanced network with much less ivy league domination. It's also important to note that for each of these inferences it is assumed that becoming a professor is the ultimate goal of the candidates. The truth of this statement will vary from department to department, but is likely most true for the History department and less true for the Computer Science and Business departments as the industry opportunities will be more tempting in those areas. To get rid of this assumption, one would have to look at the percentage of applicants accepted instead of the net number hires. There was no data cleaning needed to produce these results indicating that the validity of the published data.

Full Network Visualizations (3A)

Below we reproduce full network visualizations with the top 15% of the institutions highlighted. The size of each vertex represents the school's prestige.

```
for (dep in names(data_list)) {
  edgelist <- data_list[[dep]]$edge
  vertex <- data_list[[dep]]$vert

  # Make table that includes the weight of each edge
  weighted_edgelist <- edgelist %>%
    group_by(v, u) %>%
    summarize(count = n()) %>%
    ungroup() %>%
    left_join(vertex, by = c("v" = "u")) %>%
    select(v, u, count, institution)


  num_schools <- max(edgelist$u)

  # Make full network to make a network plot
  prestige_list <- weighted_edgelist %>%
    filter(v != num_schools, u != num_schools) %>%
    group_by(v) %>%
    summarize(
      top_school = as.double(v <= 0.15 * num_schools)[1],
      prestige = num_schools - v[1]
    ) %>%
    ungroup()


  # Set up the network to plot
  graph <- weighted_edgelist %>%
    filter(u != v, u %in% prestige_list$v, v %in% prestige_list$v) %>%
    graph_from_data_frame(directed = FALSE, vertices = prestige_list)

  plot(graph,
    vertex.size = 2 + 3 * V(graph)$top_school + V(graph)$prestige / 15,
    vertex.color = 3 + V(graph)$top_school,
    vertex.label = NA,
    main = paste(dep, "Department", sep = " ")
  )
}
```



Just like in the paper, we see that the top 15 percent of schools tend to take up central positions in the graph, which indicates they have the best faculty production and placement. There was no additional data processing needed to reproduce this result.

Faculty Placement PDFs (3B and 3C)

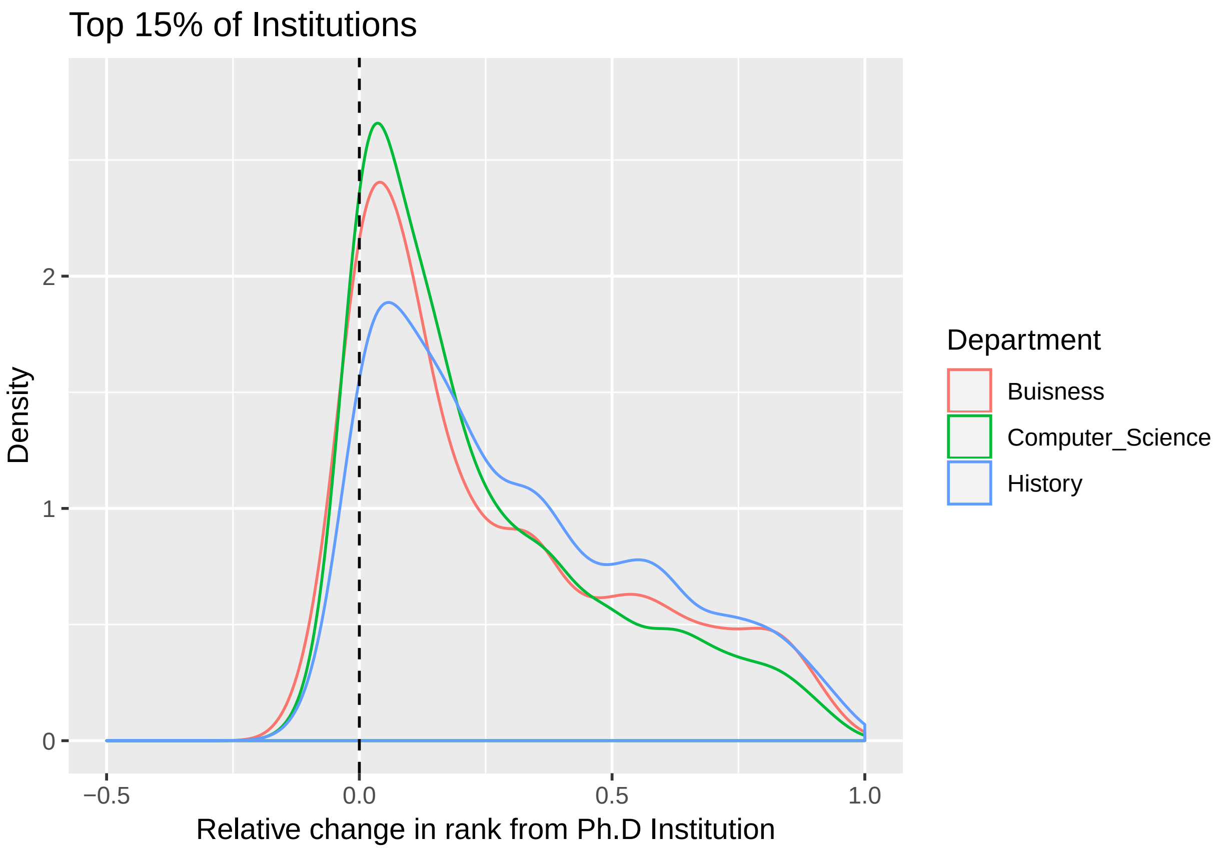
Here we reproduce the PDFs for the relative changes in rank for the top 15 schools and all other institutions.

```
# Dataframe of top 15 institutions and rest
top15 <- data.frame()
rest <- data.frame()

# All department edges and vertexes
all_edgelists <- data.frame()
all_vertexes <- data.frame()
placement_data <- data.frame()

for (dep in names(data_list)) {
  edgelist <- data_list[[dep]]$edge
  vertex <- data_list[[dep]]$vert

  num_schools <- max(edgelist$u)

  edgelist$dep <- dep
  vertex$dep <- dep

  # rbind to keep data from all the departments, adding the label of department first
  top15 <- rbind(top15, edgelist %>%
    filter(u <= .15 * num_schools) %>%
    mutate(diff = (v - u) / num_schools) %>%
    select(diff, dep, rank))

  rest <- rbind(rest, edgelist %>%
    filter(u > .15 * num_schools) %>%
```

```r
    filter(u < num_schools) %>%
    mutate(diff = (v - u) / num_schools) %>%
    select(diff, dep, rank))


  # Save all the edgelists and vertex lists into one dataframe with dep labels
  all_edgelists <- rbind(all_edgelists, edgelist)
  all_vertexes <- rbind(all_vertexes, vertex)


  # Set placement dataframes
  total_placements <- edgelist %>%
    filter(u < n()) %>%
    summarise(rows = n())
  total_placements <- as.numeric(total_placements)

  placement_data <- rbind(placement_data, edgelist %>%
    filter(u < n()) %>%
    group_by(u) %>%
    summarise(
      faculty_produced = n(),
      fraction_placements = n() / total_placements
    ) %>%
    arrange(desc(fraction_placements)) %>%
    mutate(cum_place_percent = cumsum(fraction_placements)) %>%
    mutate(fraction_schools = row_number() / n()) %>%
    mutate(dep = dep) %>%
    ungroup(edgelist))
}
```

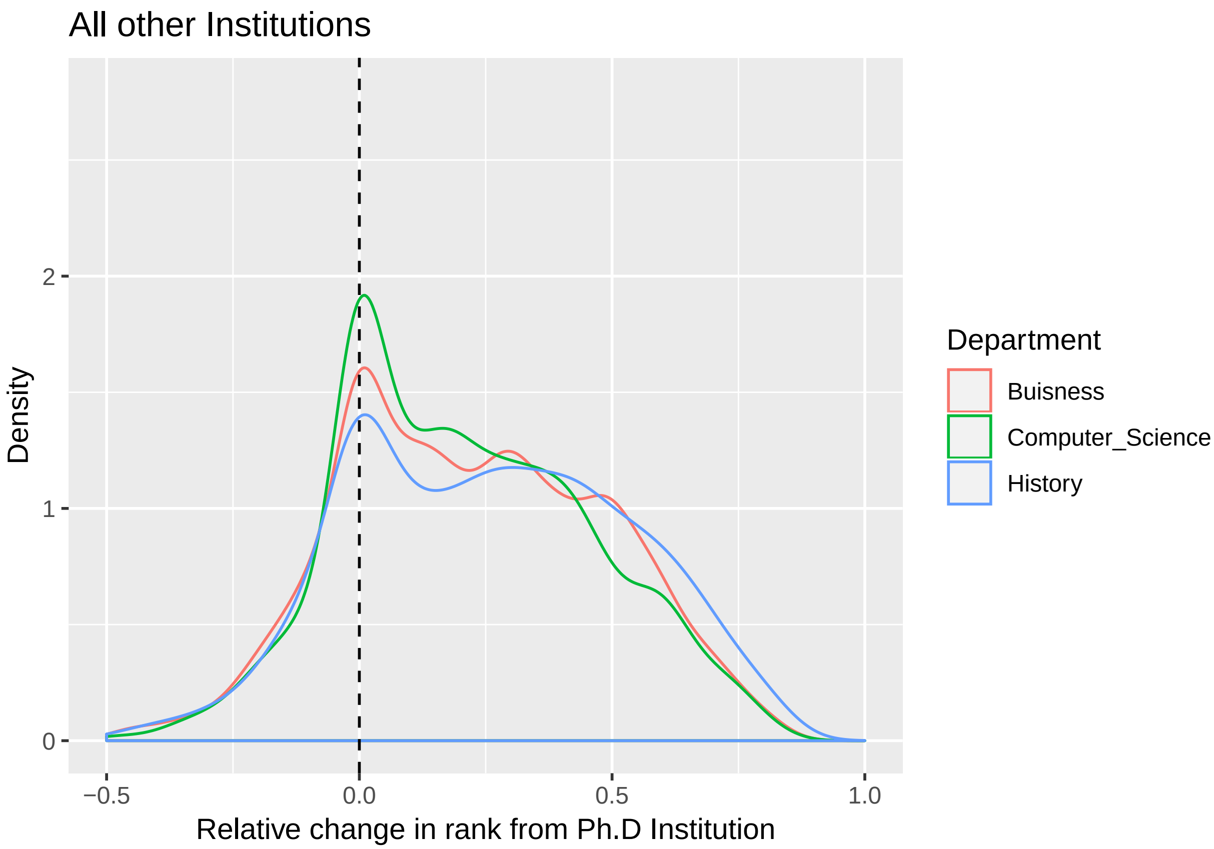
```
## Warning in `[<-.factor`(`*tmp*`, ri, value = c(1L, 12L, 14L, 1L, 9L, 4L, :
## invalid factor level, NA generated
```

```r
# Making the density plots

top15 %>%
  ggplot(aes(x = diff, color = dep)) +
  geom_density() +
  geom_vline(xintercept = 0, linetype = "dashed") +
  ylim(0, 2.8) +
  xlim(-.5, 1) +
  ggtitle("Top 15% of Institutions") +
  ylab("Density") +
  xlab("Relative change in rank from Ph.D Institution") +
  labs(color = "Department")
```




```r
rest %>%
  ggplot(aes(x = diff, color = dep)) +
  geom_density() +
  geom_vline(xintercept = 0, linetype = "dashed") +
  ylim(0, 2.8) +
  xlim(-.5, 1) +
  ggtitle("All other Institutions") +
  ylab("Density") +
  xlab("Relative change in rank from Ph.D Institution") +
  labs(color = "Department")
```

```
## Warning: Removed 36 rows containing non-finite values (stat_density).
```

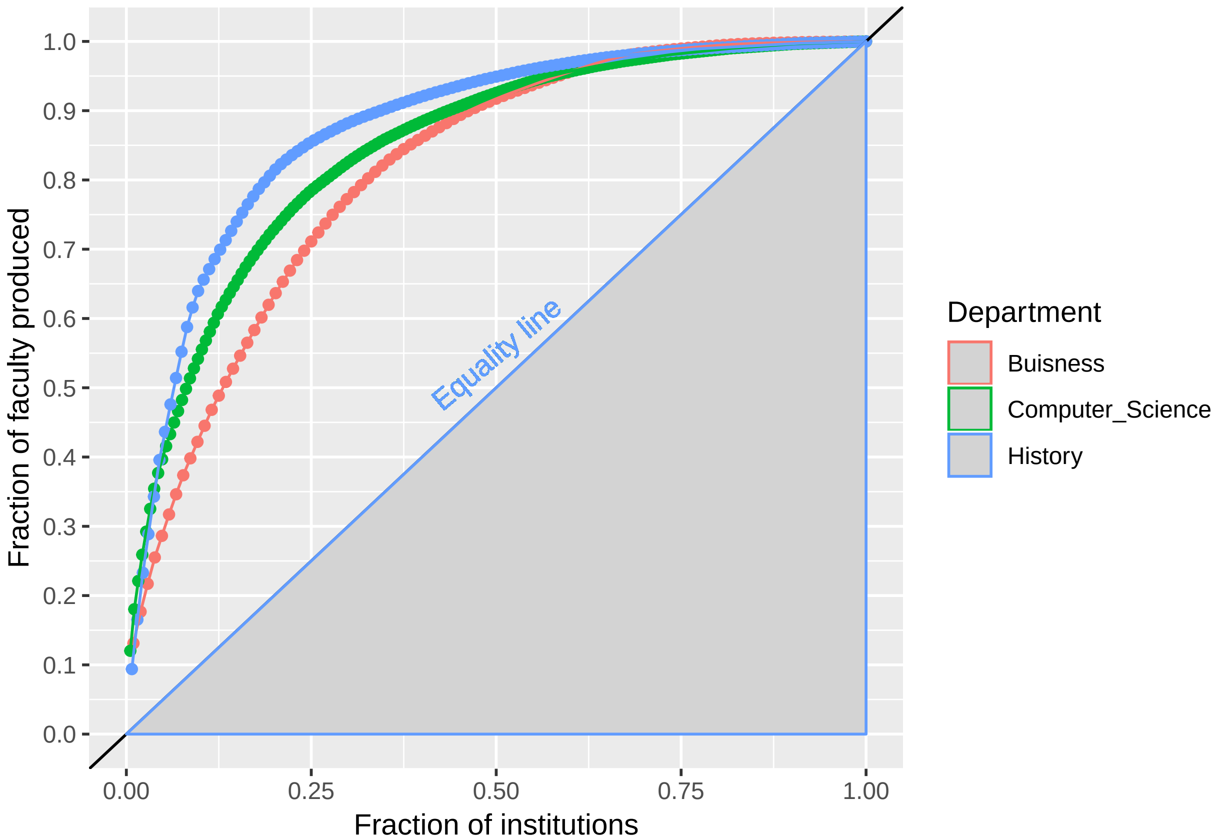


Just like in the paper, we see that candidates from the top 15% get much better placements than those in the rest of the institutions. There was no additional processing to acheive this result.

Plot Lorentz curves (Fig 2A)

Below we reproduce the Lorentz curve for the graphs. The Lorentz curve is normally used to show an income distribution, but here it is the fraction of faculty produced plotted as a function of the fraction of institutions. For this data, it shows that most of the hires come from a few select schools.

```
x <- c(0, .5, 1)
y <- c(0, .5, 1)
equality_line <- data.frame(x, y)

placement_data %>%
  ggplot(aes(x = fraction_schools, y = cum_place_percent, color = dep)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = seq(0, 1, by = 0.25)) +
  scale_y_continuous(breaks = seq(0, 1, by = 0.1)) +
  geom_abline(intercept = 0, slope = 1) +
  geom_text(aes(x = .5, y = .55, label = "Equality line", angle = 40)) +
  geom_area(data = equality_line, aes(x = x, y = y), fill = "#D3D3D3") +
  xlab("Fraction of institutions") +
  ylab("Fraction of faculty produced") +
  labs(color = "Department")
```



We see the same result as in the paper that all departments show strong inequality with History having the most and Buisness having the least. This is consistant with our network visualizations, which show the strongest ties in the History graph and the most equal ties in the Buisness graph. One possible reason for this could be that hiring methods in business rely more on the accomplishments in the world of business (a measurable statistic) while history on the other end of the spectrum does not have an easily measurable "success" metric which can be then used in faculty decisions.

Total Gini Coefficient

Here we calculte the Gini Coefficient for all departments and the full dataset. The Gini Coefficient quantifies inequality in a distribution. If the distribution is uniform then the Gini coefficient is 0, if one person owns all the wealth then it is 1.

```r
for (dep in names(data_list)) {
  edgelist <- data_list[[dep]]$edge

  # Finding the gini coefficient for each department, using the library reldist
  school_counts <- edgelist %>%
    filter(v != num_schools) %>%
    group_by(v) %>%
    summarize(counts = n()) %>%
    ungroup()

  # Here the coefficients look very small when looking at it split by department
  G <- gini(school_counts$counts, runif(n = nrow(school_counts)))
```

```
  cat(dep, "Gini Coefficient: ", G, "\n")
}
```

```
## Buisness Gini Coefficient:  0.2397501
## Computer_Science Gini Coefficient:  0.3097472
## History Gini Coefficient:  0.2821473
```

```
school_counts <- all_edgelists %>%
  left_join(all_vertexes, by = c("v" = "u", "dep" = "dep")) %>%
  select(v, u, institution) %>%
  filter(institution != "All others") %>%
  group_by(institution) %>%
  summarize(counts = n()) %>%
  ungroup()

# Here the coefficients look very small when looking at it split by department
G <- gini(school_counts$counts)

cat("Whole Dataset Coefficient:", G)
```

```
## Whole Dataset Coefficient: 0.4686504
```

We were unable to reproduce the Gini Coefficients reported in the paper. They report `G=0.62-0.76` indicating extremely strong inequality. Our values are much lower on the whole, but still indicate inequality.

Extension: Predicting Hiring Party Ranks and Prestige with Different Factors

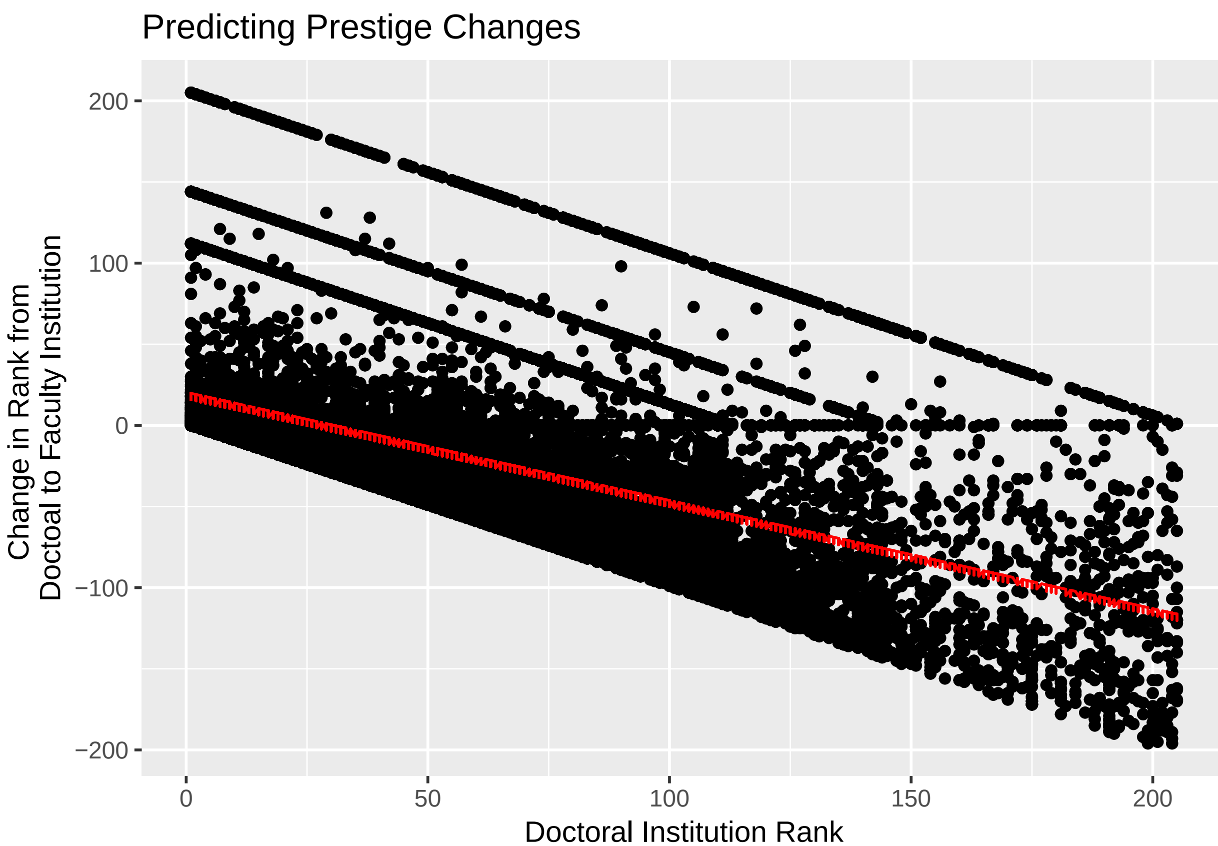
Most of the results in the paper are descriptive, they plot some aspect of the graph and make the argument for inequality. For our extension, we aim to augment these results by introducing some more predictive results. Specifically, we use simple linear regression models to predict different properties of the hiring party. While we do not expect these simple models to be performant at predicting the future because the underlying distribution is expected to change in time, careful analysis of their coefficients will allow us to peel off general insights and trends for this particular dataset.

Predicting rank of the hiring party from gender

We begin by attempting to predict the rank of the hiring party from the gender of the candidate. This will allow us to draw some insights about the effect of gender on faculty hiring.

```
regress <- all_edgelists %>%
  mutate(y = u - v) %>%
  left_join(all_vertexes, by = c("v" = "u", "dep" = "dep")) %>%
  filter(institution != "All others") %>%
  mutate(num_gender = gender == "F") %>%
  select(y, v, num_gender)


# I could do train/test split, but the model's not very good, and we're really just doing thsi to inter
model <- lm(y ~ (v + num_gender), data = regress)
regress$pred <- predict(model, regress)

regress %>%
```

```r
  ggplot(aes(x = v, y = y)) +
  geom_point() +
  geom_line(aes(y = pred), color = "red") +
  ylab("Change in Rank from
Doctoal to Faculty Institution") +
  xlab("Doctoral Institution Rank") +
  ggtitle("Predicting Prestige Changes")
```




```r
summary(model)
```

```
##
## Call:
## lm(formula = y ~ (v + num_gender), data = regress)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -83.725 -26.054 -15.387   7.889 188.557
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    20.386789   0.606901  33.592  < 2e-16 ***
## v              -0.666643   0.007396 -90.136  < 2e-16 ***
## num_genderTRUE -3.944168   0.799277  -4.935  8.1e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.27 on 18565 degrees of freedom
```

```
## Multiple R-squared:  0.3049, Adjusted R-squared:  0.3048
## F-statistic:  4071 on 2 and 18565 DF,  p-value: < 2.2e-16
```

We can see that the model predicts women to go to higher prestige schools relative to their doctoral school (the difference is statistically significant), although only by about 4 ranks, so not a very big difference. We can also see that as people go to lower prestige schools, the model predicts they will go to higher prestige schools. Of course, this doesn't really tell the full story, since by looking at the graph we can that there are more points for the higher prestige doctoral schools (x close to 1), and this model obviously doesn't capture the people who attended s these schools but didn't go on to become faculty professors.

Predicting Hiring School Prestige and Rank from Doctoral Rank and Gender

Next, we predict the prestige of the hiring school from both doctoral rank and gender. This is a simple extension of the previous regression and will allow us to draw insites about

```r
# testing what how much prestige points yield the best, how to weigh k

for (k in c(0, 0.5, 1, 2, 4, 6, 8, 10, 50)) {
  cat("
k =", k)

  rank_regress <- all_edgelists %>%
    mutate(asst = as.double(rank == "Asst")) %>%
    mutate(full = as.double(rank == "Full")) %>%
    mutate(assoc = as.double(rank == "Assoc")) %>%
    mutate(y = (u - v) - k * (3 * full + 2 * assoc + asst)) %>%
    left_join(all_vertexes, by = c("v" = "u", "dep" = "dep")) %>%
    filter(institution != "All others") %>%
    mutate(num_gender = gender == "F") %>%
    select(y, v, num_gender)


  # No train test split as we want coef rather than predictions
  model <- lm(y ~ (v + num_gender), data = rank_regress)


  cat("
R squared:", summary(model)$r.squared)

  # We can see that scaling the change in prestige by the faculty rank doesn't help the predictive powe
}
```

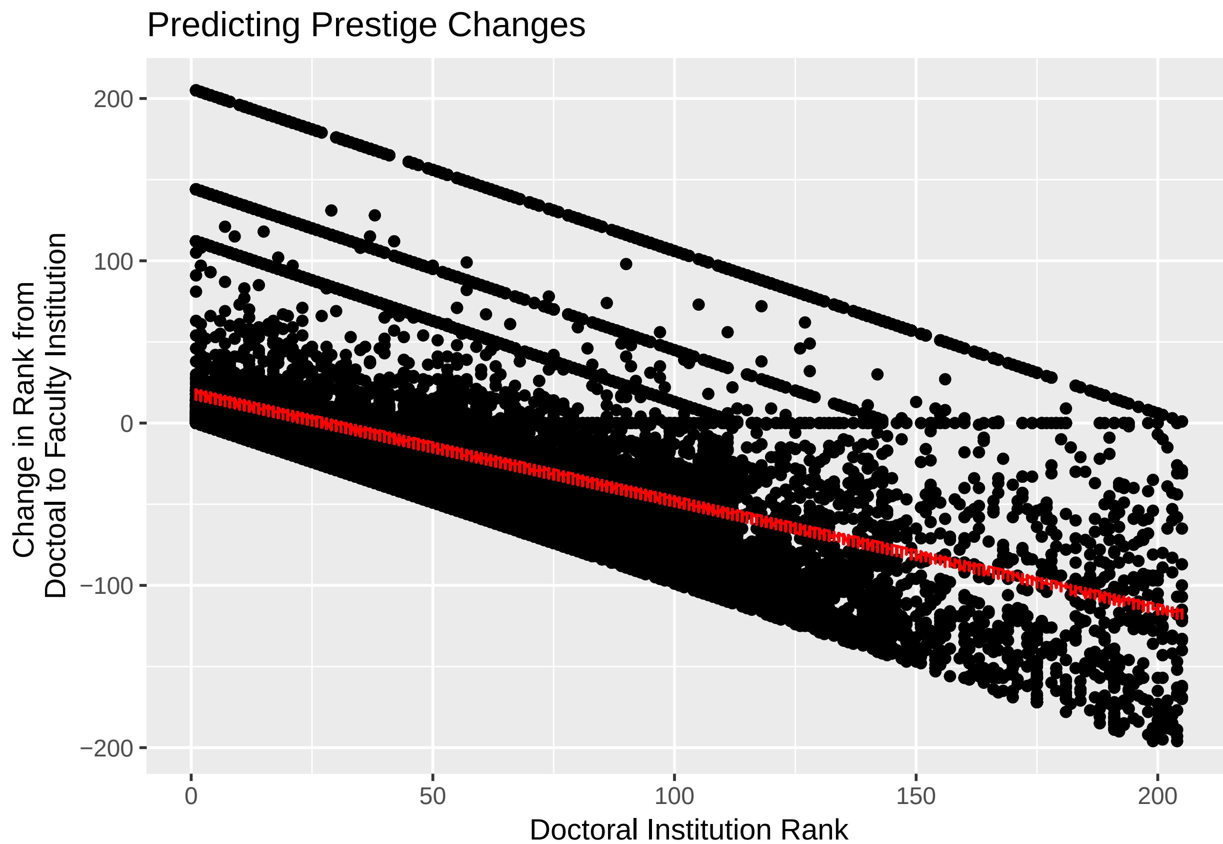
```
##
## k = 0
## R squared: 0.3048886
## k = 0.5
## R squared: 0.3045917
## k = 1
## R squared: 0.3042663
## k = 2
## R squared: 0.3035302
## k = 4
## R squared: 0.3017194
## k = 6
## R squared: 0.2994642
```

```
## k = 8
## R squared: 0.296777
## k = 10
## R squared: 0.2936746
## k = 50
## R squared: 0.1873009
```

Using faculty ranks as predictors

Finally, we add faculty ranks as predictors for the model.

```r
rank_predictors <-
  all_edgelists %>%
    mutate(asst = as.double(rank == "Asst")) %>%
    mutate(full = as.double(rank == "Full")) %>%
    mutate(assoc = as.double(rank == "Assoc")) %>%
    mutate(y = (u - v)) %>%
    left_join(all_vertexes, by = c("v" = "u", "dep" = "dep")) %>%
    filter(institution != "All others") %>%
    mutate(num_gender = gender == "F") %>%
    select(y, v, num_gender, full, assoc)


model <- lm(y ~ v + num_gender + assoc + full, data = rank_predictors)
rank_predictors$pred <- predict(model, rank_predictors)

rank_predictors %>%
  ggplot(aes(x = v, y = y)) +
  geom_point() +
  geom_line(aes(y = pred), color = "red") +
  ylab("Change in Rank from
Doctoal to Faculty Institution") +
  xlab("Doctoral Institution Rank") +
  ggtitle("Predicting Prestige Changes")
```




```r
cat("
coefficients:
")
```

```
##
## coefficients:
```

```r
summary(model)
```

```
##
## Call:
## lm(formula = y ~ v + num_gender + assoc + full, data = rank_predictors)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -84.861 -25.914 -15.178   7.731 189.736
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    18.816342   0.895067  21.022  < 2e-16 ***
## v              -0.665045   0.007461 -89.132  < 2e-16 ***
## num_genderTRUE -3.557435   0.810498  -4.389 1.14e-05 ***
## assoc           0.875597   0.943187   0.928  0.35324
## full            2.388616   0.885013   2.699  0.00696 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.26 on 18563 degrees of freedom
```

```
## Multiple R-squared:  0.3052, Adjusted R-squared:  0.305
## F-statistic:  2038 on 4 and 18563 DF,  p-value: < 2.2e-16
```

```r
cat("
R squared:", summary(model)$r.squared)
```

```
##
## R squared: 0.3051953
```

We can see that faculty with a rank of full have a slightly bigger difference in prestige, going to less prestigious schools, although its such a small difference I don't believe it means very much. For that coefficient we do see a small p-value, and it makes intuitive sense that it is easier to get a full faculty position at a lower prestige school. Still, the difference is not nearly as big as I would have expected. As for Associate and Assistants, we see no statistically significant difference between the two coefficients.

Conclusions

"FODJFIODSFISDIFOSDIOFJIO"

The following is a list of all packages used to generate these results. (Leave at very end of file.)

```r
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.4
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] here_0.1        reldist_1.6-6   igraph_1.2.4    modelr_0.1.3
##  [5] forcats_0.3.0   stringr_1.4.0   dplyr_0.8.0.1   purrr_0.3.0
##  [9] readr_1.3.1     tidyr_0.8.2     tibble_2.0.1    ggplot2_3.1.0
## [13] tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.0          lubridate_1.7.4     lattice_0.20-38
##  [4] rprojroot_1.3-2     assertthat_0.2.0    digest_0.6.18
##  [7] R6_2.4.0            cellranger_1.1.0    plyr_1.8.4
## [10] backports_1.1.3     acepack_1.4.1       evaluate_0.13
## [13] httr_1.4.0          pillar_1.3.1        rlang_0.3.1
## [16] lazyeval_0.2.1      readxl_1.3.0        data.table_1.12.2
## [19] rstudioapi_0.9.0    rpart_4.1-13        Matrix_1.2-15
## [22] checkmate_1.9.3     rmarkdown_1.11      labeling_0.3
## [25] splines_3.5.2       foreign_0.8-71      htmlwidgets_1.3
## [28] munsell_0.5.0       broom_0.5.1         compiler_3.5.2
## [31] xfun_0.4            pkgconfig_2.0.2     base64enc_0.1-3
```

```
## [34] mgcv_1.8-26         htmltools_0.3.6     nnet_7.3-12
## [37] tidyselect_0.2.5    gridExtra_2.3       htmlTable_1.13.1
## [40] Hmisc_4.2-0         crayon_1.3.4        withr_2.1.2
## [43] grid_3.5.2          nlme_3.1-137        jsonlite_1.6
## [46] gtable_0.2.0        magrittr_1.5        scales_1.0.0
## [49] cli_1.1.0           stringi_1.3.1       latticeExtra_0.6-28
## [52] xml2_1.2.0          generics_0.0.2      Formula_1.2-3
## [55] RColorBrewer_1.1-2  tools_3.5.2         glue_1.3.0
## [58] hms_0.4.2           survival_2.43-3     yaml_2.2.0
## [61] colorspace_1.4-0    cluster_2.0.7-1     rvest_0.3.2
## [64] knitr_1.21          haven_2.0.0
```