Performance of linking researchers to theses

Flavio & Christoph

13 September, 2022

Contents

Overview . . . 1
Linking scores . . . 1
Link performance by graduation year . . . 2
Notes . . . 4
Number of linked advisors . . . 5
Compare number of links across iterations within fields . . . 6
Check overlap of institution names and years . . . 8
Note: the "usable" links are saved to the db in src/dataprep/main/link/prep_linked_data.py . . . 9
Fraction with matched advisor status by cohort of own graduation . . . 9

This script makes some plots of the advisor links and saves the most plausible links to a table in the database.

```
# parameters for selecting links
min_score_advisors <- 0.7 # minimum score from dedupe
max_year_diff <- 5 # maximum difference between advisory and own publication at institution. 5 is arbit
max_uniname_distance <- 0.02 # keep only links where the jarowinkler distance between the institution n
```

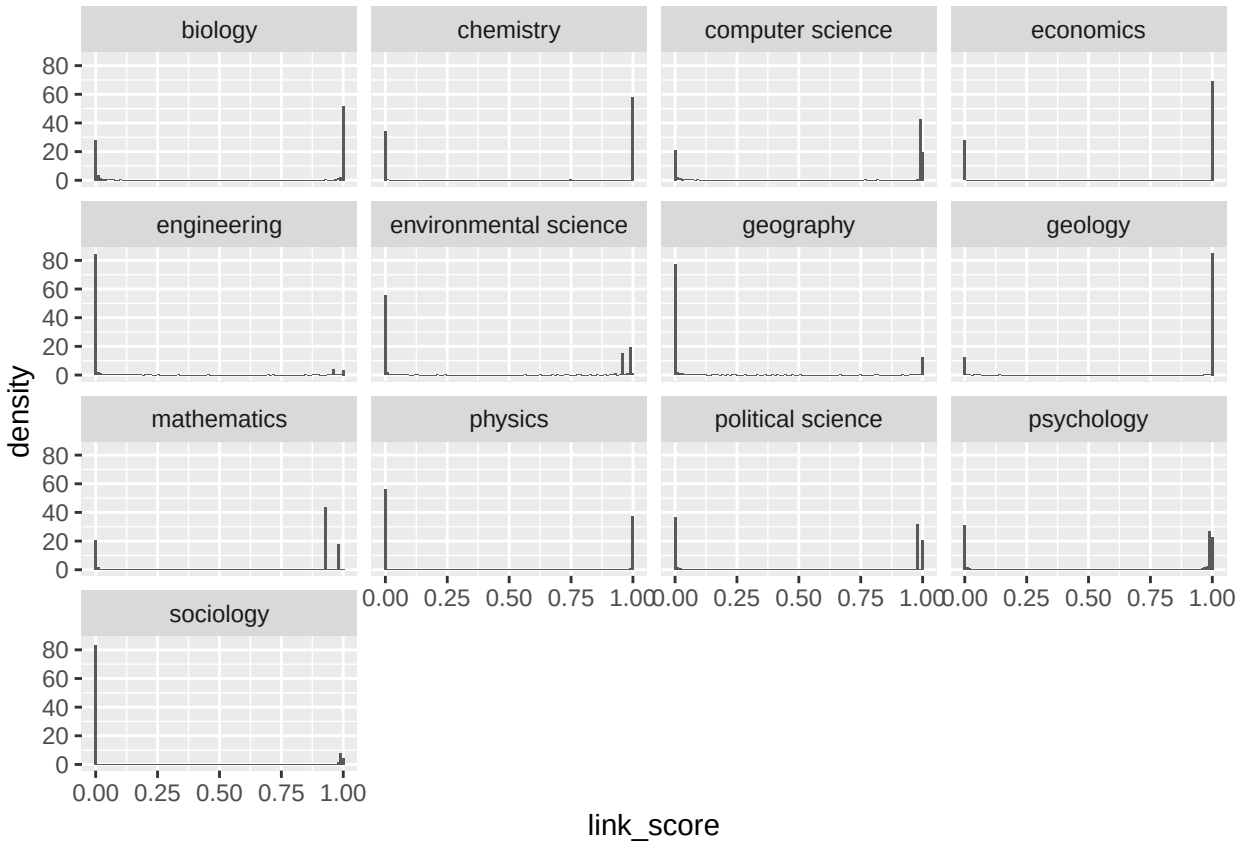
Overview

```
current_links <- collect(current_links)
linked_advisors <- collect(linked_advisors)
theses <- collect(theses)
authors_affiliation <- collect(authors_affiliation)
linking_info <- collect(linking_info)
pq_fields_mag <- collect(pq_fields_mag)
```

Linking scores

- conditioning on link score > 0.7 is fine

```
linked_advisors %>%
  left_join(linking_info, by = "iteration_id") %>%
  ggplot(aes(x = link_score)) +
  geom_histogram(bins = 100, aes( y = ..density..)) +
  facet_wrap(~field)
```



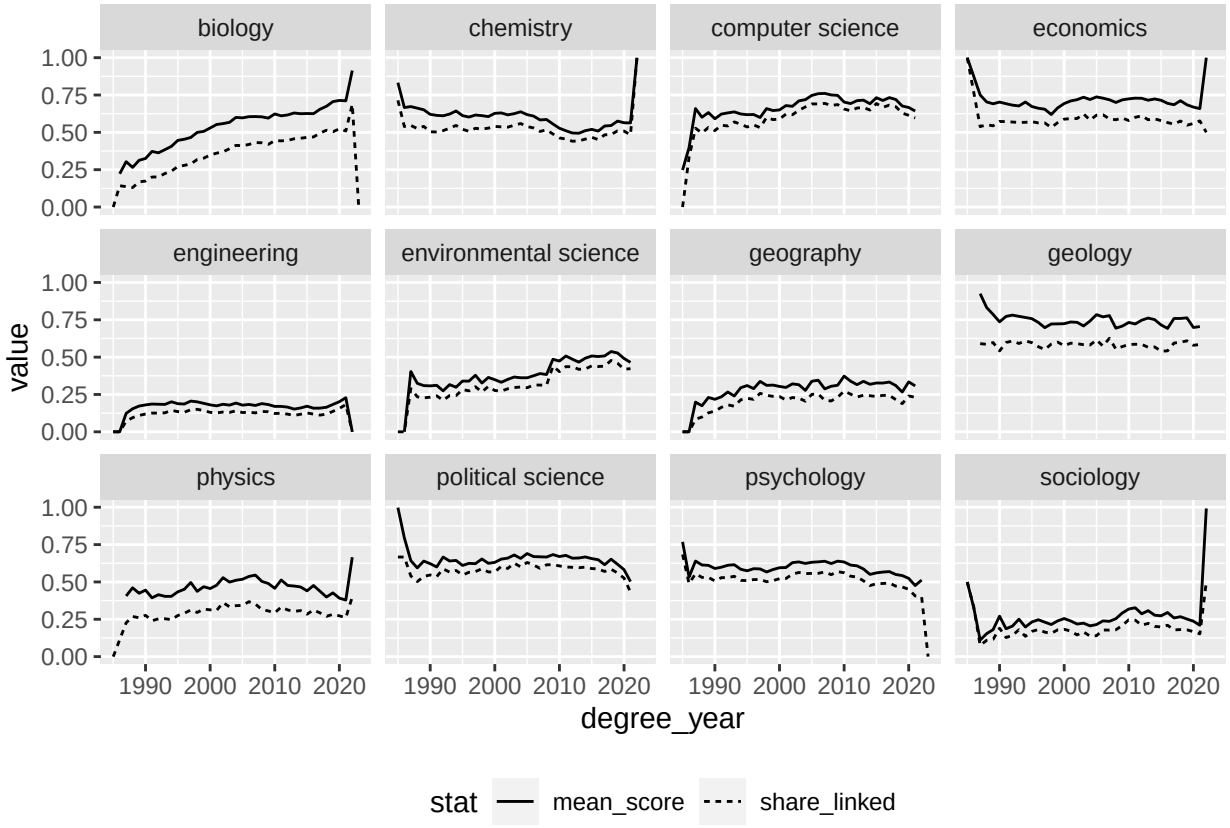
Link performance by graduation year

- fraction of listed advisors where the link_score is above the treshold
- the mean link score for advisors where dedupe finds a link (link_score is not NA)
- NOTE: the field here is assigned based on the first reported in the dissertation, and the crosswalked to the MAG field
  - in the figure above, we used the field from iteration_id, but this only works for advisors that dedupe suggests to be a link

```
keep_fields <- c("biology", "chemistry", "computer science",
                 "economics", "engineering", "environmental science",
                 "geography", "geology", "mathetmatics", "physics",
                 "political science", "psychology", "sociology")

score_by_year <- theses %>%
  filter(degree_year >= 1985) %>%
  left_join(linked_advisors,
            by = "relationship_id") %>%
  left_join(pq_fields_mag, by = "goid") %>%
  filter(field %in% keep_fields)

# %>%
#   left_join(linking_info,
#             by = "iteration_id")

score_by_year %>%
  mutate(link_score_adj = ifelse(is.na(link_score), -1, link_score)) %>%
```

```
  group_by(degree_year, field) %>%
  summarise(mean_score = mean(link_score, na.rm = TRUE),
            #p50_score = quantile(link_score, probs = 0.5),
            share_linked = mean(link_score_adj > min_score_advisors),
            .groups = "drop") %>%
  pivot_longer(cols = all_of(c("mean_score", "share_linked")),
               names_to = "stat") %>%
  ggplot(aes(x = degree_year, y = value)) +
  geom_line(aes(linetype = stat)) +
  facet_wrap(~field) +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
score_by_year %>%
  filter(field == "biology") %>%
  filter(is.na(link_score)) %>%
  filter(degree_year < 2000) %>%
  group_by(firstname, lastname, uni_name, degree_year) %>%
  summarise(nb = n(),
            .groups = "drop") %>%
  arrange(desc(nb)) %>%
  head(10)
```

```
## # A tibble: 10 x 5
##    firstname  lastname  uni_name                              degree_year    nb
##    <chr>      <chr>     <chr>                                       <int> <int>
##  1 john a     gerlt     university of maryland college park          1994     6
```

```
##  2 asim       dasgupta  university of california los angeles        1990     5
##  3 barry s    cooperman university of pennsylvania                  1996     5
##  4 douglas e  eveleigh  rutgers university                         1993     5
##  5 michael    freeling  university of california berkeley         1997     5
##  6 mingdaw    tsai      ohio state university                     1997     5
##  7 peter      elsbach   new york university                        1999     5
##  8 c brent    theurer   university of arizona                     1997     4
##  9 c channa   reddy     pennsylvania state university             1995     4
## 10 chawnshang chang     university of wisconsin madison           1996     4
```

```
# score_by_year %>% filter(lastname == "dasgupta" & firstname == "asim" & !is.na(iteration_id)) # never
# score_by_year %>% filter(lastname == "freeling" & firstname == "michael") # never linked

# scale this up? check all the main fields of the authors with such names? -- tedious
```

Notes

- Reasons for why advisor not linked
  - they are not sampled for linking either in the mag or proquest data
    - most plausibly because they are assigned to different fields
  - institution names do not overlap
  - dedupe does not find a link even though it should
    - but how can it explain the time trend?
- Comparing fields in MAG and ProQuest dissertations
  - General
    - not linking an advisor in biology does not mean do not link them in chemistry if the thesis is also classified in chemistry
    - in the data above, this happens if biology is listed at position 0
  - Biology
    - main field chemistry: gerlt, cooperman, eveleigh (two of them with long careers, but both in chemistry), tsai
    - main field biology: dasgupta, freeling
    - at least one of the dissertations of freeling are sampled for the linking
  - Sociology
    - different main field: ishisaka, coulton (medicine), howell (geography), mindel (psychology)
    - not in MAG, but findable on google: khleif, gullerud
    - not in MAG, not findable on google: liff
- Next steps
  - widen the sampled field in MAG
  - re-train and re-check

Here is some python code to look at the learned settings, based on

- https://github.com/dedupeio/rlr/blob/master/rlr/lr.py (new dedupe does not use this anymore I think)
- https://github.com/dedupeio/dedupe/blob/5742efc7fc696c06d3327e038541532e584551a8/dedupe/api.py

The predicates are similar for all three fields I looked at. I do not know how the weights correspond to the logit regression coefficients

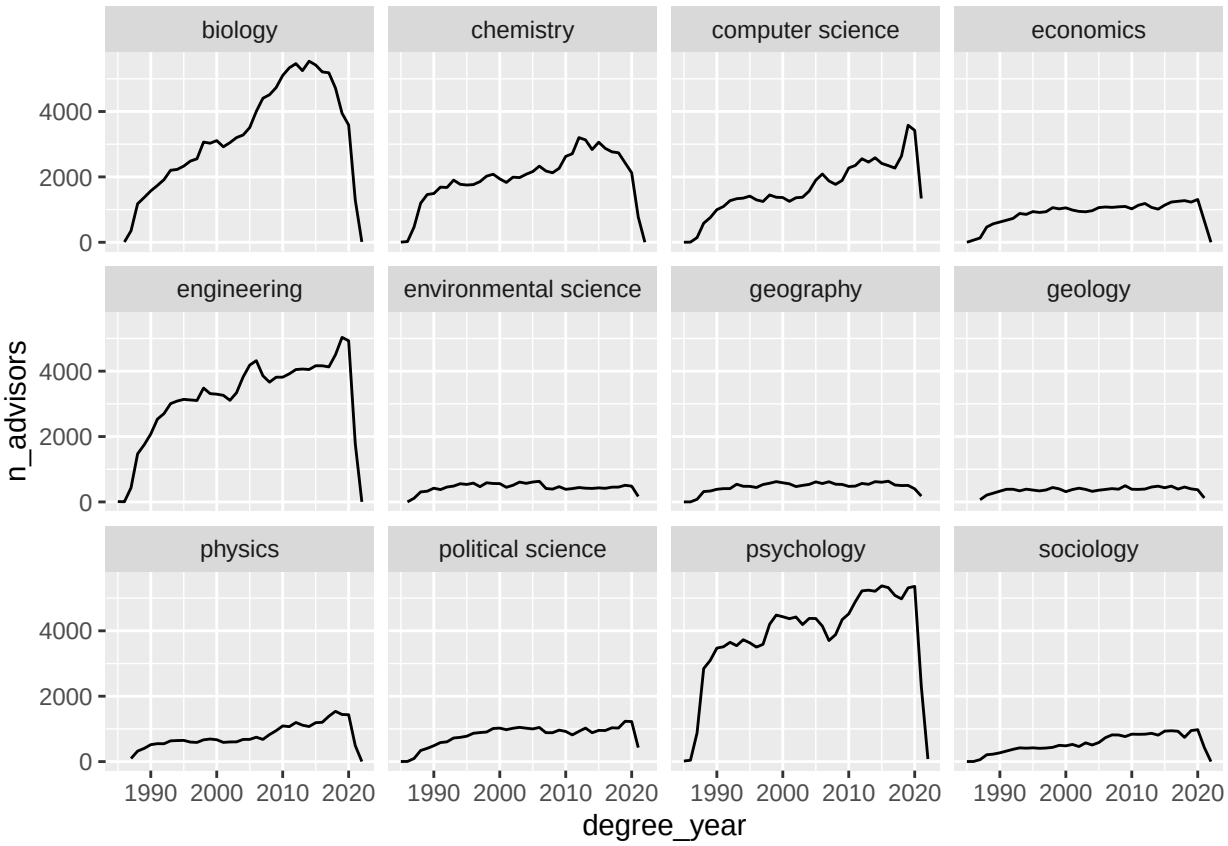
```
sf_biology = "/mnt/ssd/DedupeFiles/advisors/settings_biology_1985_2022_institutionTrue_fieldofstudy_cat
sf_chemistry = "/mnt/ssd/DedupeFiles/advisors/settings_chemistry_1985_2022_institutionTrue_fieldofstudy
sf_cs = "/mnt/ssd/DedupeFiles/advisors/settings_computer_science_1985_2022_institutionTrue_fieldofstudy

with open(sf_biology, "rb") as sf:
```

```
    linker_biology = dedupe.StaticRecordLink(sf)

with open(sf_chemistry, "rb") as sf:
    linker_chemistry = dedupe.StaticRecordLink(sf)


with open(sf_cs, "rb") as sf:
    linker_cs = dedupe.StaticRecordLink(sf)

linker_biology.predicates
linker_chemistry.predicates
linker_cs.predicates

linker_biology.classifier.weights
linker_chemistry.classifier.weights
linker_cs.classifier.weights
```

Number of linked advisors

- not sure this is still relevant?

```
score_by_year %>%
  filter(!is.na(link_score)
         & field %in% keep_fields) %>%
  group_by(degree_year, field) %>%
  summarise(n_advisors = n(),
            .groups = "drop") %>%
  ggplot(aes(x = degree_year, y = n_advisors)) +
  geom_line() +
  facet_wrap(~field)
```



old comments

- for instance, a student of michael j lambert (authorid 2120159045; relationship id 303670971_0 in proquest) from pre-1990 is link score of 0.02, but should be a clear link

Compare number of links across iterations within fields

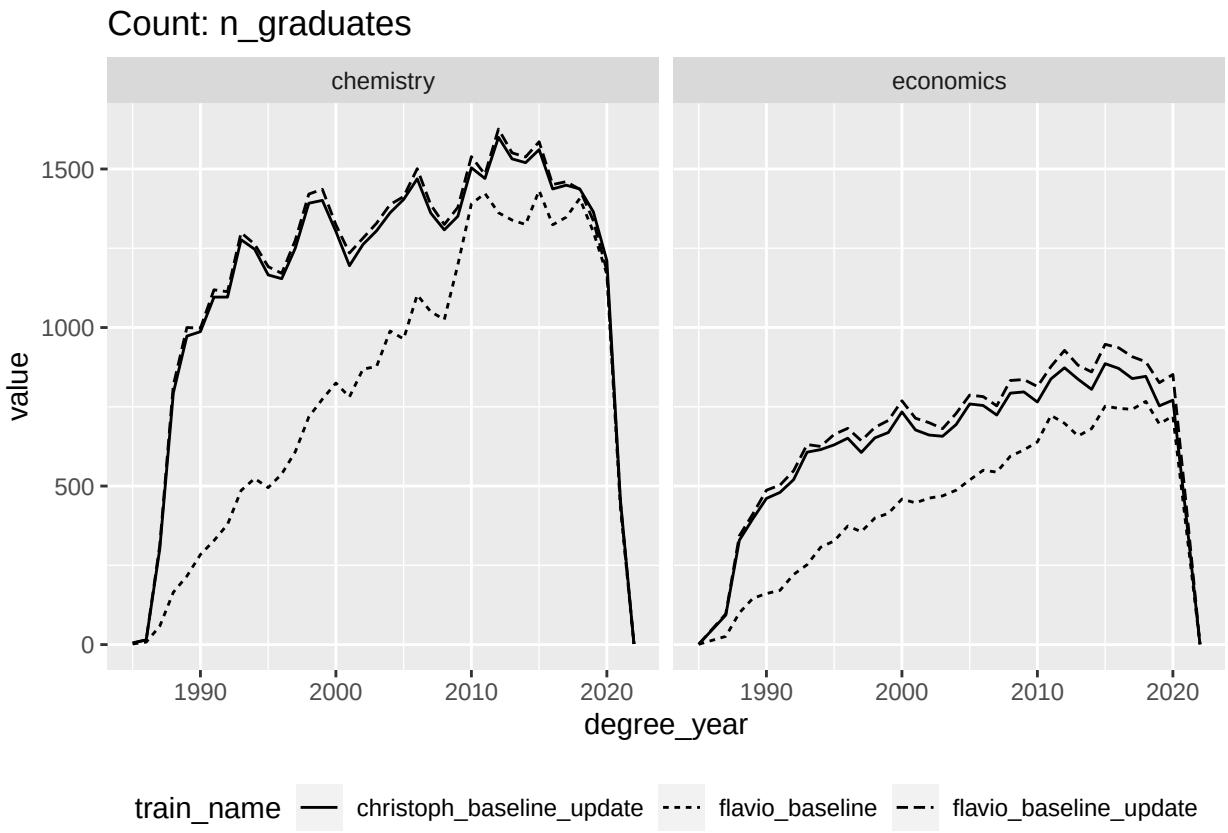
```
fields_iter_compare <- c("economics", "chemistry")
min_score <- 0.8

keep_iter_ids <- tbl(con, "linking_info_advisors") %>%
  filter(field %in% fields_iter_compare) %>%
  filter(testing == 0) %>%
  collect() %>%
  group_by(field, train_name) %>%
  arrange(iteration_id) %>%
  mutate(nb = n(),
         id = row_number()) %>%
  ungroup() %>%
  filter(id == nb) %>%
  select(iteration_id, field, train_name)

linked_ids_to_compare <- tbl(con, "linked_ids_advisors") %>%
  inner_join(
    tbl(con, "linking_info_advisors") %>%
      filter(field %in% fields_iter_compare),
    by = "iteration_id"
```

```
  ) %>%
  inner_join(
    tbl(con, "pq_advisors") %>%
      select(relationship_id, goid),
    by = "relationship_id"
  ) %>%
  inner_join(
    tbl(con, "pq_authors") %>%
      select(goid, degree_year),
    by = "goid"
  ) %>%
  collect() %>%
  filter(iteration_id %in% keep_iter_ids$iteration_id)
```

Number of graduates with at least 1 advisor

```
d_sum <- linked_ids_to_compare %>%
  filter(link_score >= min_score) %>%
  group_by(train_name, field, degree_year) %>%
  summarise(n_advisors = n(),
            n_graduates = n_distinct(goid),
            .groups = "drop") %>%
  pivot_longer(cols = starts_with("n_"), names_to = "variable")

plotvars <- c("n_graduates")

map(.x = plotvars,
    .f = ~d_sum %>%
      filter(variable == .x) %>%
      ggplot(aes(x = degree_year, y = value)) +
      geom_line(aes(linetype = train_name)) +
      facet_wrap(~field) +
      theme(legend.position = "bottom") +
      labs(title = paste0("Count: ", .x))
    )
```

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



Check overlap of institution names and years

```
d_main <- linked_advisors %>%
  filter(link_score > min_score_advisors) %>%
  left_join(theses %>%
              mutate(fullname = paste0(firstname, " ", lastname)) %>%
              select(relationship_id, degree_year, uni_name, fullname),
            by = "relationship_id") %>%
  inner_join(current_links %>%
               select(author_name, AuthorId),
             by = "AuthorId") %>%
  # join on year; filter on max similarity within relationship_id; still need to examine multiple match
  left_join(authors_affiliation,
            by = c("AuthorId")) %>%
  mutate(dist_uni_name = stringdist(uni_name, affil_name, method = "jw"),
         dist_year = abs(degree_year - Year)) %>%
  group_by(relationship_id) %>%
  filter(dist_uni_name == min(dist_uni_name)) %>%
  filter(dist_year == min(dist_year)) %>%
  mutate(nb = n()) %>% # can still have multiple links if e.g. the dissertation is in x, but the affili
  ungroup()

d_main <- d_main %>%
  filter(!duplicated(relationship_id))

cat("Split of links by whether years are >/<", max_year_diff, "apart")
```

```
## Split of links by whether years are >/< 5 apart
table(d_main$dist_year <= max_year_diff)
```

```
##
## FALSE  TRUE
##  2188 73641
```

```
d_main <- d_main %>%
  filter(dist_year <= max_year_diff & dist_uni_name <= max_uniname_distance)
```

Note: the "usable" links are saved to the db in src/dataprep/main/link/prep_linked_data.py

Fraction with matched advisor status by cohort of own graduation

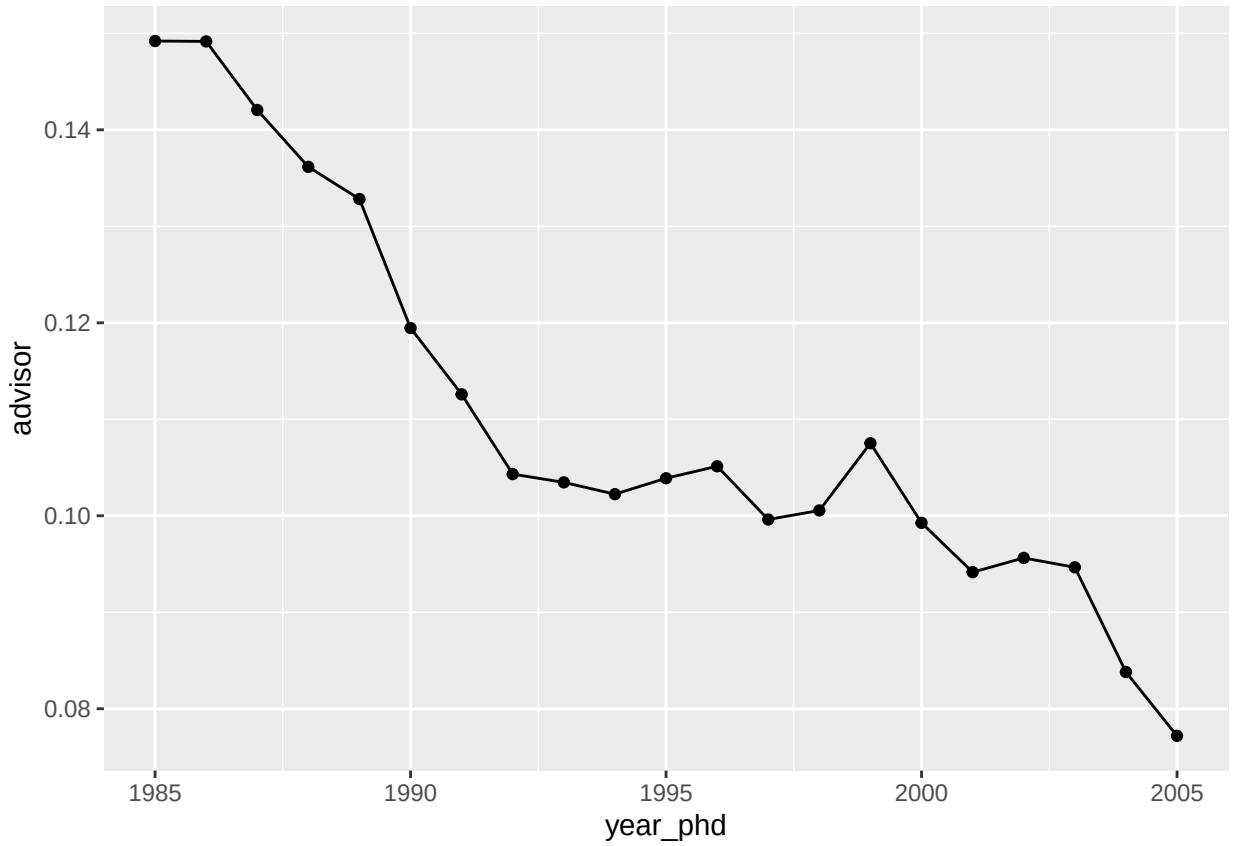
```
pq_authors <- tbl(con, "pq_authors") %>% collect()
```

```
d_agg <- d_main %>%
  select(AuthorId, relationship_id, degree_year) %>%
  group_by(AuthorId) %>%
  filter(degree_year == min(degree_year)) %>%
  ungroup() %>%
  rename(year_firstadvisee = degree_year) %>%
  filter(!duplicated(AuthorId))
```

```
d_links <- current_links %>%
  left_join(pq_authors %>%
              select(goid, year_phd = degree_year),
            by = "goid") %>%
  left_join(d_agg, by = "AuthorId")
```

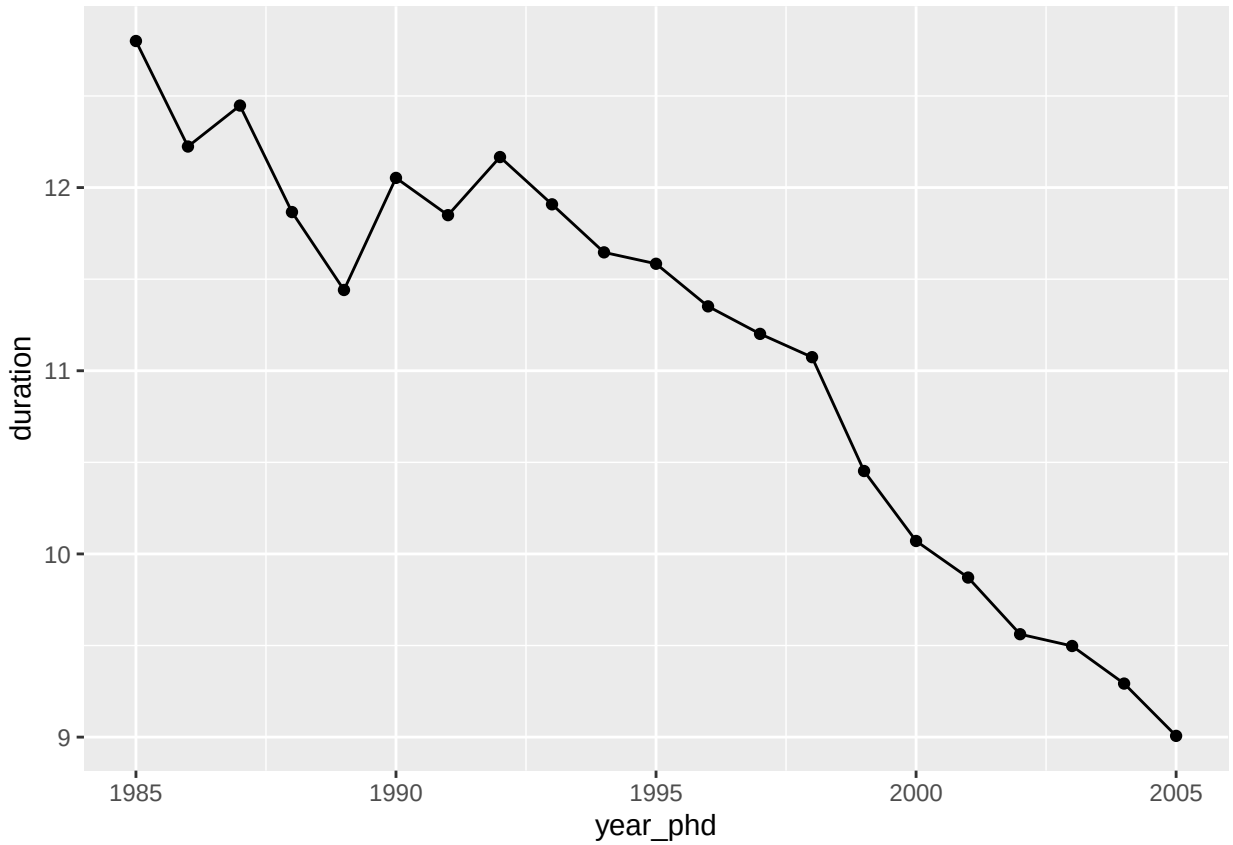
Fraction of authors that eventually becomes advisor

```
d_links %>%
  mutate(advisor = ifelse(is.na(year_firstadvisee), 0, 1)) %>%
  group_by(year_phd) %>%
  summarise(advisor = mean(advisor),
            .groups = "drop") %>%
  ggplot(aes(x = year_phd, y = advisor)) +
  geom_point() +
  geom_line()
```



Duration to advisor

```
d_links %>%
  filter(!is.na(year_firstadvisee)) %>%
  mutate(duration = year_firstadvisee - year_phd) %>%
  group_by(year_phd) %>%
  summarise(duration = mean(duration),
            .groups = "drop") %>%
  ggplot(aes(x = year_phd, y = duration)) +
  geom_point() +
  geom_line()
```


duration
12
11
10
9
1985
1990
1995
2000
2005
year_phd