

# Performance of linking researchers to theses

Flavio & Christoph

21 September, 2022

## Contents

Overview . . . . .	1
Linking scores . . . . .	1
Average link score by graduation year . . . . .	2
Compare number of links across iterations within fields . . . . .	4
Check overlap of institution names and years . . . . .	6
Note: the “usable” links are saved to the db in src/dataprep/main/link/prep_linked_data.py . . .	7
Fraction with matched advisor status by cohort of own graduation . . . . .	7

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
```

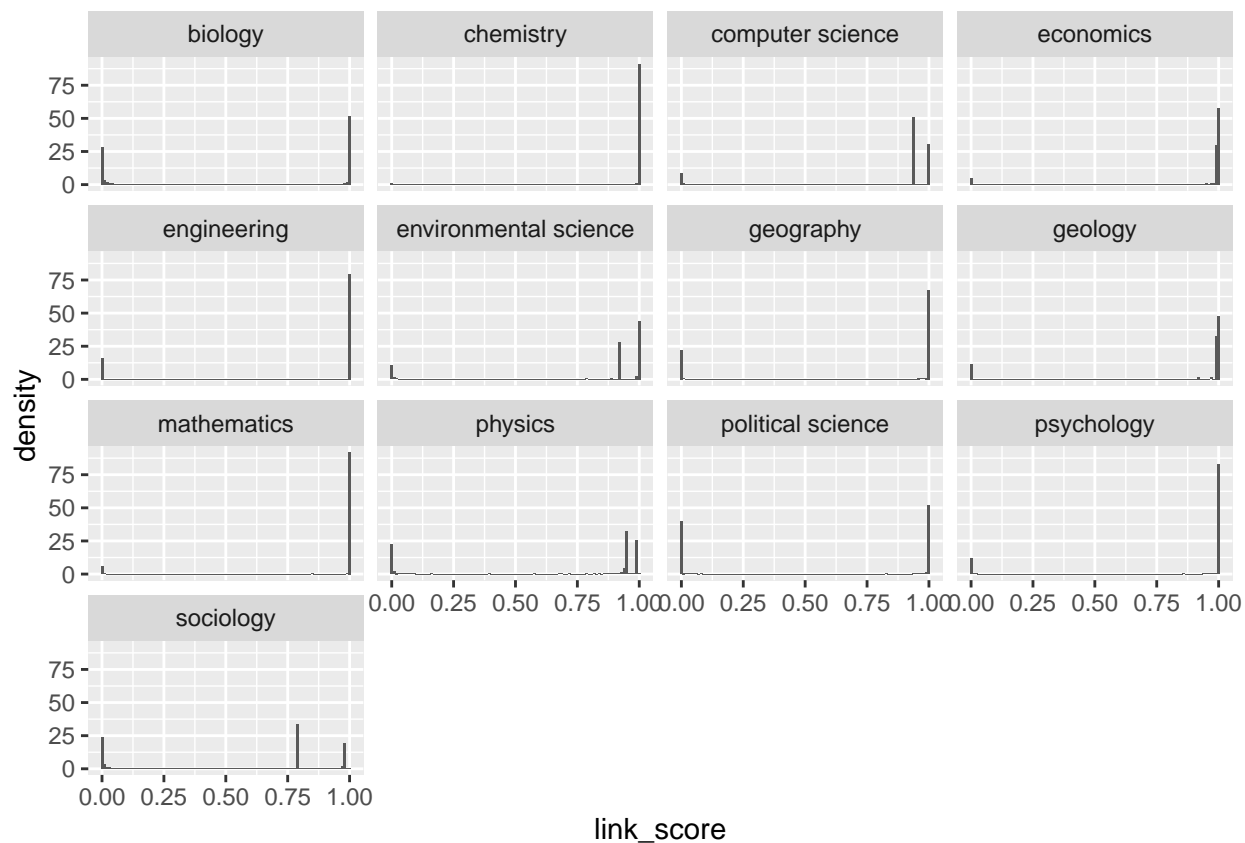
## Overview

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

## Linking scores

- conditioning on link score > 0.7 is fine

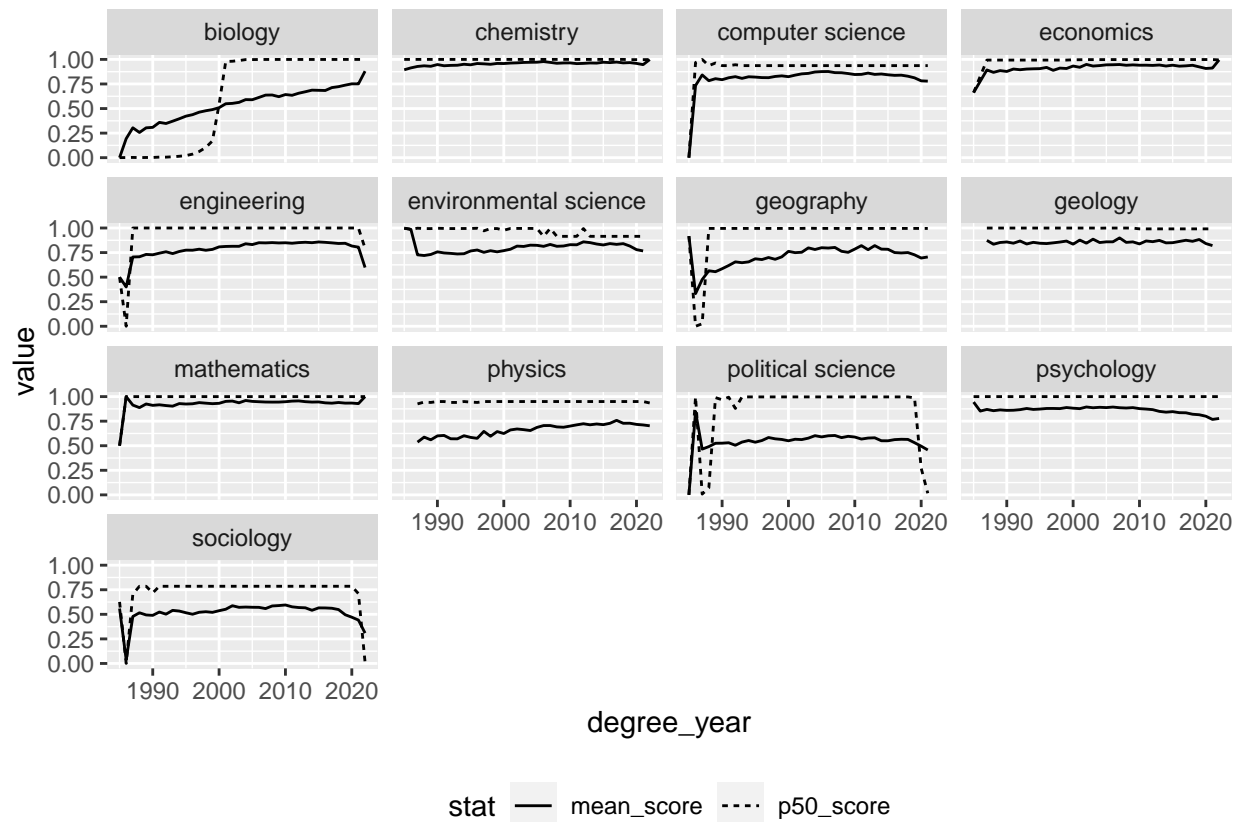
```
linked_advisors %>%
  left_join(linking_info, by = "iteration_id") %>%
  ggplot(aes(x = link_score)) +
  geom_histogram(bins = 100, aes(y = ..density..)) +
  facet_wrap(~field)
```



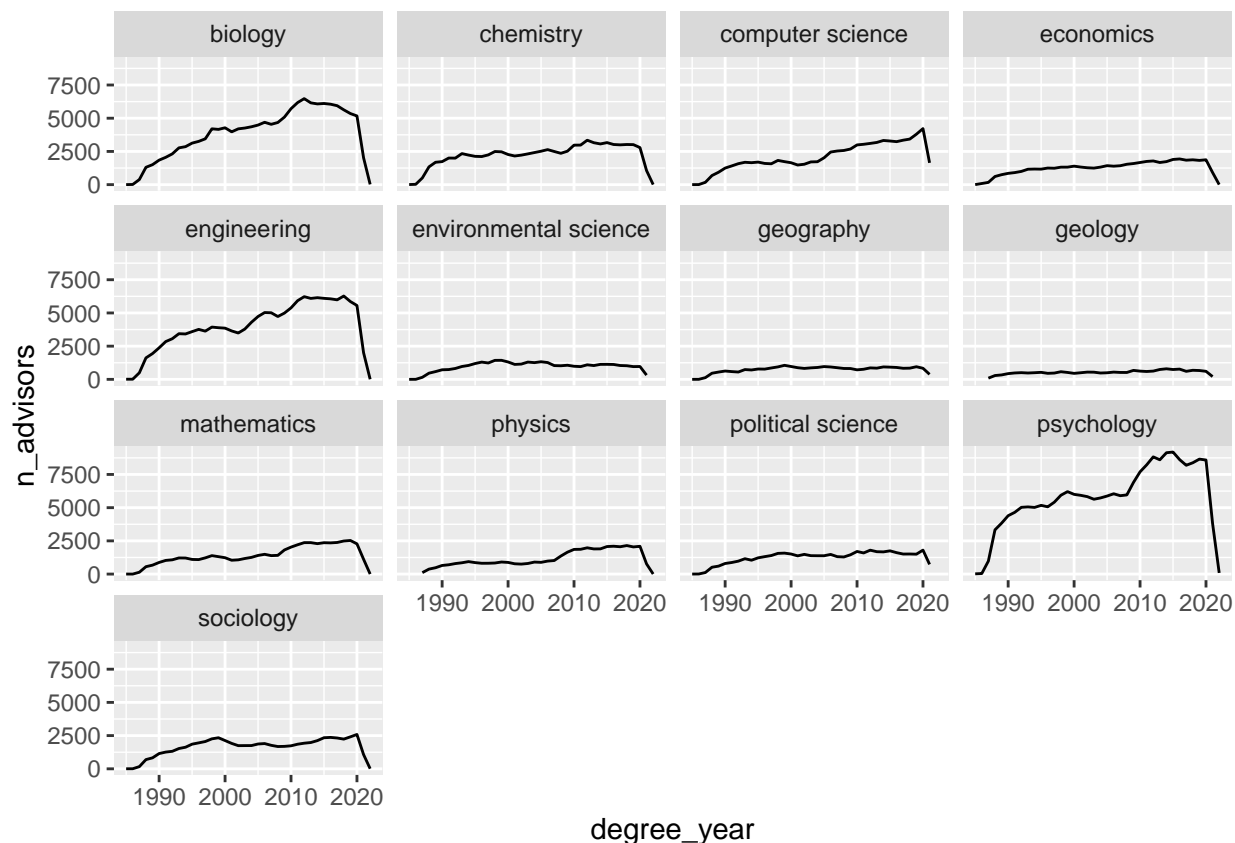
### Average link score by graduation year

```
score_by_year <- linked_advisors %>%
  left_join(theses %>%
    select(degree_year, relationship_id),
    by = "relationship_id") %>%
  left_join(linking_info,
    by = "iteration_id")

score_by_year %>%
  group_by(degree_year, field) %>%
  summarise(mean_score = mean(link_score),
    p50_score = quantile(link_score, probs = 0.5),
    .groups = "drop") %>%
  pivot_longer(cols = ends_with("_score"),
    names_to = "stat") %>%
  ggplot(aes(x = degree_year, y = value)) +
  geom_line(aes(linetype = stat)) +
  facet_wrap(~field) +
  theme(legend.position = "bottom")
```



```
score_by_year %>%
  group_by(degree_year, field) %>%
  summarise(n_advisors = n(),
            .groups = "drop") %>%
  ggplot(aes(x = degree_year, y = n_advisors)) +
  geom_line() +
  facet_wrap(~field)
```



```
# another stat
# linked advisors per linked dissertation
# versus advisors per dissertation (in the US)

# score_by_year %>%
#   mutate(grp = ifelse(degree_year < 2000, "pre", "post")) %>%
#   ggplot(aes(x = link_score)) +
#   geom_histogram() +
#   facet_wrap(~grp)
```

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")

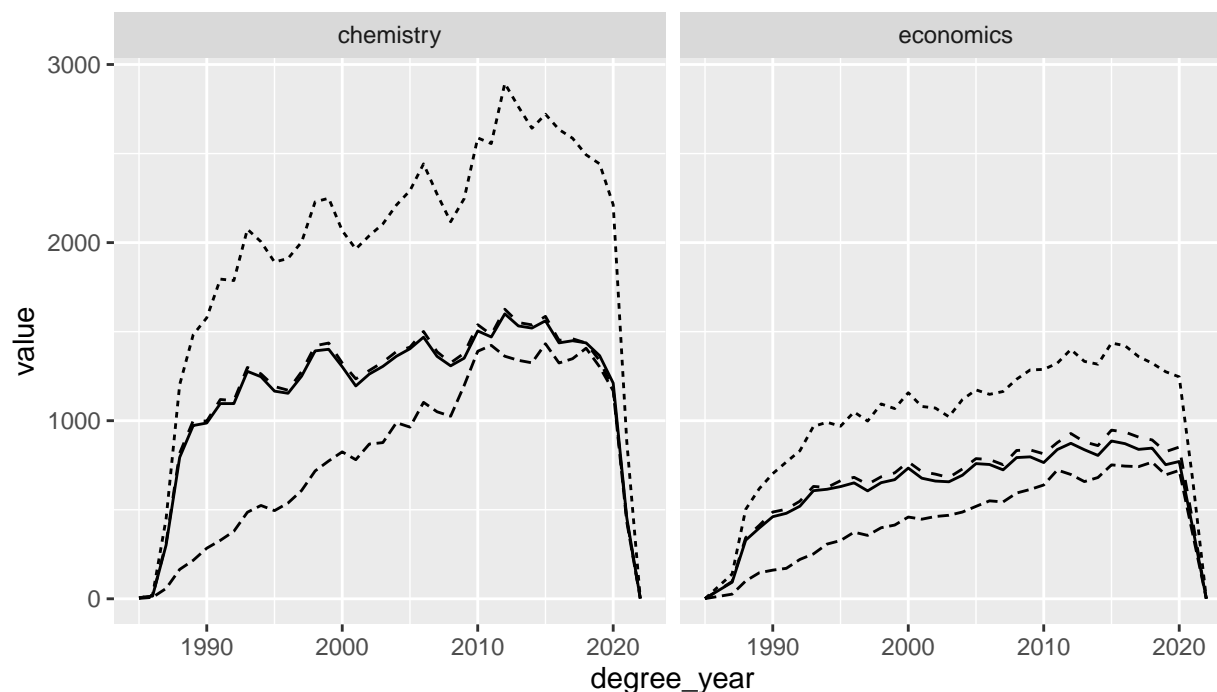
plotvars <- c("n_graduates")

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]]
```

Count: n\_graduates



train\_name — christoph\_baseline\_update ..... christoph\_degree0 --- flavio\_baseline - - flavio\_degree0

## 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 matches
  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 affiliation is in y
  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
## 4084 80552

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")

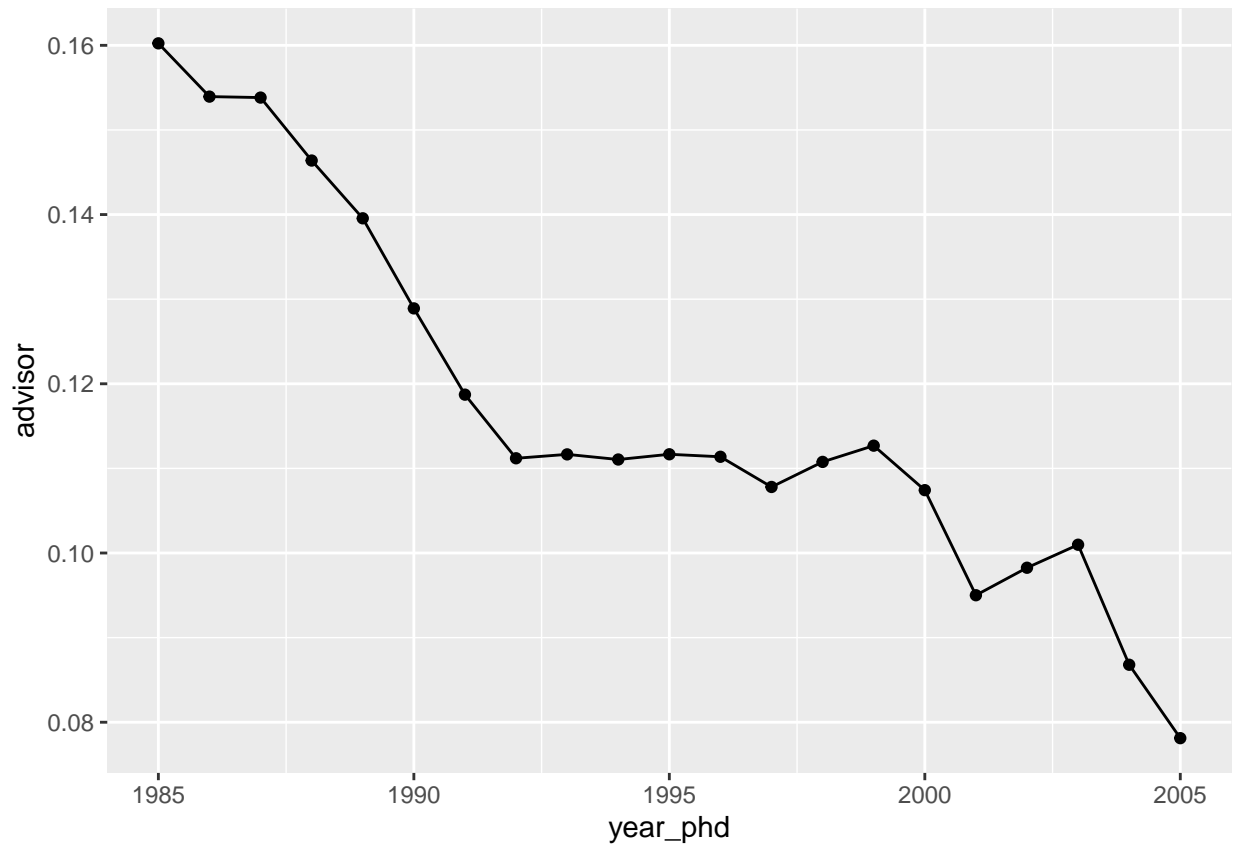
```

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()
```



