Standardisation

Thomas Klebel

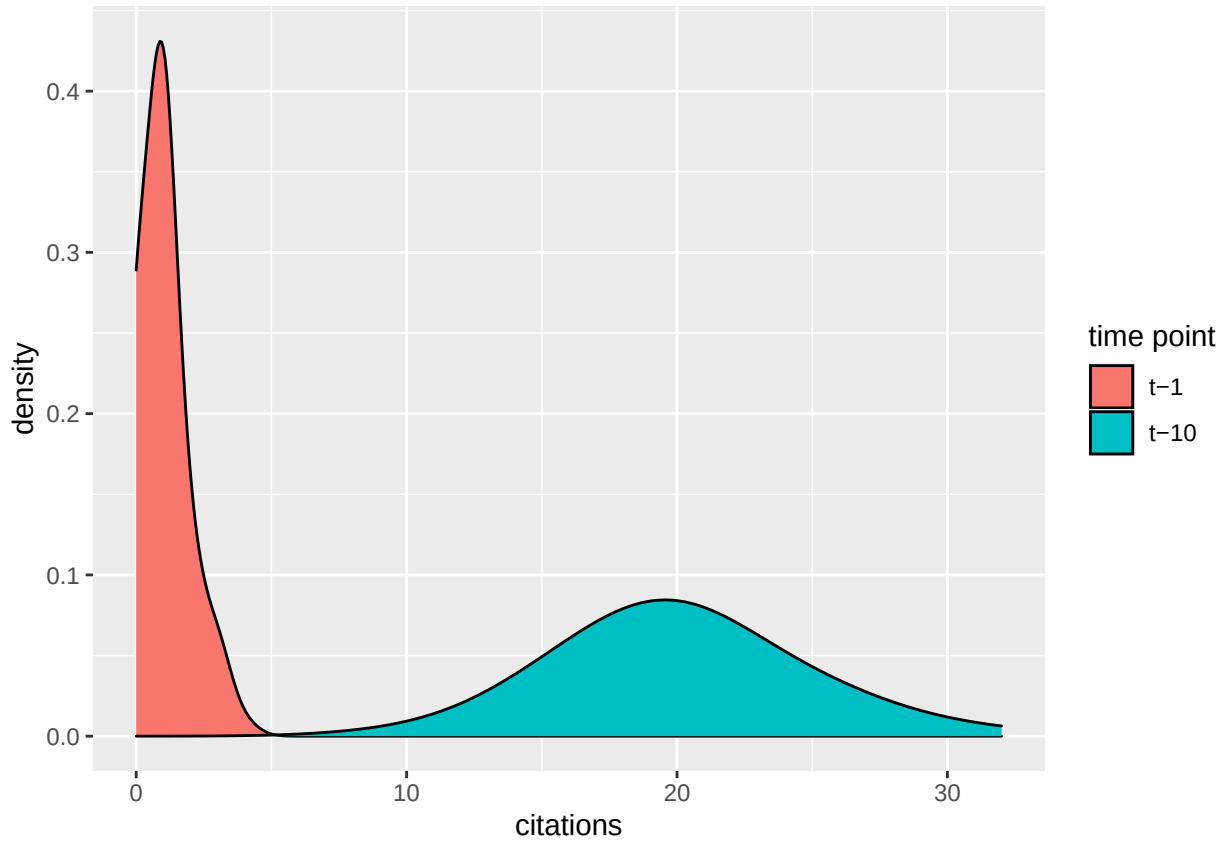
27 9 2021

First, we sample two independent vectors, which could represent citations at time t_{-1} and t_{-10}, i.e. citations for articles published in 2019 and in 2010. We assume a poisson distribution, with an average of one citation for papers from 2019 and an average of 20 citations for papers from 2010.

We assume that there is only one journal.

```
set.seed(9823)
df <- tibble(`t-1` = rpois(100, 1),
             `t-10` = rpois(100, 20))

df %>%
  pivot_longer(everything()) %>%
  ggplot(aes(value, fill = name)) +
  geom_density(adjust = 2) +
  labs(x = "citations", fill = "time point")
```



To show how the standardisation works, we go through several steps:

1. We transform for long format for easier manipulation
2. We set up further calculations separately for t_{-1} and t_{-10}
3. We calculate the mean citations for each year (for t_{-1} and t_{-10} separately)
4. We standardise individual values by dividing through the journal mean.

```
# simply transform the data for easier manipulation
df_long <- df %>%
  pivot_longer(everything())

standardised_vals <- df_long %>%
  group_by(name) %>%
  mutate(mean_val = mean(value),
         std_val = value/mean_val)

standardised_vals %>%
  head(10) %>%
  knitr::kable()
```

name	value	mean_val	std_val
t-1	2	0.99	2.0202020
t-10	15	20.07	0.7473842
t-1	1	0.99	1.0101010
t-10	20	20.07	0.9965122
t-1	0	0.99	0.0000000
t-10	18	20.07	0.8968610
t-1	1	0.99	1.0101010
t-10	14	20.07	0.6975585
t-1	1	0.99	1.0101010
t-10	21	20.07	1.0463378

In the last step, we aggregate the data by using the mean. As we can see, there is no difference in the standardised means, regardless of how long citations had time to accumulate.

```
standardised_vals %>%
  summarise(means = mean(std_val)) %>%
  knitr::kable()
```

name	means
t-1	1
t-10	1

However, if we shift the normalising factor by e.g. subtracting 5, we artificially create a more "impactful" sample, since now our papers have more citations than the average in that year. This is reflected in the subsequent means.

```r
df %>%
  pivot_longer(everything()) %>%
  group_by(name) %>%
  mutate(mean_val = mean(value),
         # subtract 5 if we are at time-point t-10, i.e. in 2010
         mean_val = case_when(name == "t-10" ~ mean_val - 5,
                              TRUE ~ mean_val),
         std_val = value/mean_val) %>%
  summarise(means = mean(std_val)) %>%
  knitr::kable()
```

name	means
t-1	1.000000
t-10	1.331785