Title Goes Here

Jane Doe1,2,✉, John Q. Doe1, Peder Ås1, Juan Pérez3, and Max Mustermann

April 17, 2024

Abstract

Text of abstract

1 Formatting Open Science Group  
2 Federation of Planets  
3 Acme Corporation

✉ Correspondence: [Jane Doe <janedoe@fosg.org>](mailto:janedoe@fosg.org)

Keywords: keyword 1; keyword 2; keyword 3

Highlights: These are the highlights.

# Introduction

Here is a citation (Marwick, 2017)

# Background

# Methods

# Results

library(tidyverse)  
library(here)  
library(ggbeeswarm)  
# This CSV file was downloaded from our data sheet here  
# https://docs.google.com/spreadsheets/d/1Jwe3UqJyedrV-QWlwR\_44\_\_t4xBVrCfxGyhXdi3E0sg/edit?resourcekey#gid=1686084773  
# note that you may need to download it again to get the latest updates!  
  
jobdata <- read\_csv(here::here('analysis/data/raw\_data/Tenure Track Job Advertisements in Archaeology (Responses) - Form Responses 1.csv')) %>%   
 # simplify the column names   
 janitor::clean\_names()  
  
total\_number\_of\_ads\_in\_our\_sample <- nrow(jobdata) # 550

We have 609 job advertisements in our sample

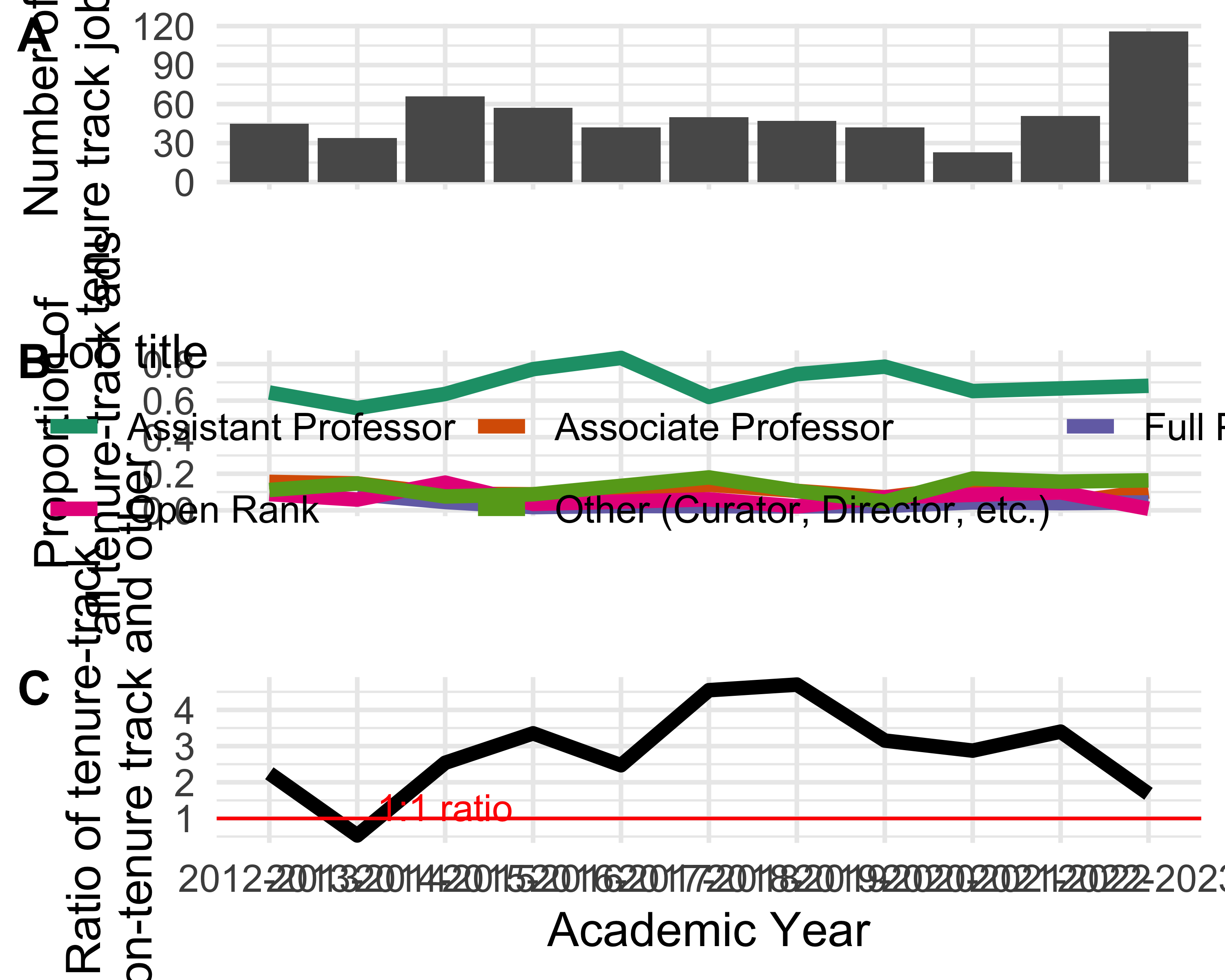
# we can get the year from the URL to the Academic Job Ads Wiki  
  
year\_ad\_posted <-   
jobdata %>%   
 pull(url\_to\_data\_source\_e\_g\_paste\_in\_url\_to\_the\_jobs\_wiki\_page) %>%   
 str\_extract(., "[[0-9]]{4}-[[0-9]]{4}|2021-22") %>%   
 str\_replace("2021-22", "2021-2022")  
 # fix for 2021-22 DONE!  
 # fix for 2023 DONE!  
  
jobdata <-   
 jobdata %>%   
 mutate(year\_ad\_posted = year\_ad\_posted) %>%   
 drop\_na(year\_ad\_posted)  
  
fig\_how\_many\_jobs\_per\_year <-   
ggplot(jobdata) +  
 aes(year\_ad\_posted) +  
 geom\_bar() +  
 scale\_x\_discrete(name = "") +  
 ylab("Number of\ntenure track job ads") +  
 theme\_minimal(base\_size = 14) +  
 guides(x = "none")

**?@fig-how-many-jobs-per-year** shows how many jobs per year in our sample

# how many jobs of each rank per year?  
  
jobdata <-  
jobdata %>%  
 # simplify rank descriptions  
 mutate(title\_of\_position\_tenure\_track\_jobs\_only = tolower(title\_of\_position\_tenure\_track\_jobs\_only)) %>%  
 mutate(job\_title\_simple = case\_when(  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,  
 "assistant prof|asst. prof|asst prof") ~ "Assistant Professor",  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,  
 "associate prof|assoc. prof") ~ "Associate Professor",  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,  
 "full prof") ~ "Full Professor",  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,  
 "assistant or associate prof|assistant/associate prof") ~ "Assistant or Associate Professor",  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,  
 "open rank|open-rank|assistant, associate, or full prof|assistant prof, associate prof, or prof") ~ "Open Rank",  
 .default = "Other (Curator, Director, etc.)"))  
  
# explore over time  
fig\_prop\_by\_job\_title\_per\_year <-   
jobdata %>%  
 group\_by(year\_ad\_posted) %>%  
 count(job\_title\_simple) %>%  
 mutate(prop = n / sum(n)) %>%  
 ggplot() +  
 aes(year\_ad\_posted,  
 prop,  
 group = job\_title\_simple,  
 colour = job\_title\_simple) +  
 geom\_line(linewidth = 2) +  
 theme\_minimal(base\_size = 14) +  
 xlab("") +  
 ylab("Proportion of\nall tenure-track ads") +  
 theme(legend.position = c(0.5, 0.5)) +  
 scale\_colour\_brewer(palette = "Dark2") +  
 guides(colour = guide\_legend(nrow=2,  
 byrow=TRUE,  
 "Job title")) +  
 guides(x = "none")

# ratio of tenure-track to untenured positions  
# base URL changes after 2018\_2019  
  
base\_url\_to\_2019 <- "https://academicjobs.fandom.com/wiki/Archaeology\_Jobs\_"  
base\_url\_after\_2020 <- "https://academicjobs.fandom.com/wiki/Archaeology\_"  
  
# starts at 2010-2011  
# fix for 2021-22  
# base UR  
  
years\_to\_2019 <- map\_chr(2012:2019, ~str\_glue('{.x}-{.x +1}'))  
years\_after\_2020 <- map\_chr(2020:2022, ~str\_glue('{.x}-{.x +1}'))  
# though it seems to start at 2007-8: https://academicjobs.fandom.com/wiki/Archaeology\_07-08  
  
# make a set of URLs for each page for each year  
urls\_for\_each\_year <- c(str\_glue('{base\_url\_to\_2019}{years\_to\_2019}'),   
 str\_glue('{base\_url\_after\_2020}{years\_after\_2020}')) %>%   
 str\_replace("2021-2022", "2021-22")  
  
library(rvest)  
  
# all years  
urls\_for\_each\_year\_headers <-   
map(urls\_for\_each\_year,  
 ~.x %>%   
 read\_html() %>%   
 html\_nodes('.mw-headline') %>%   
 html\_text())  
  
# keep only headings that are actual jobs, they include the terms:  
job\_headings <- c("college", "university")  
  
total\_number\_of\_jobs\_per\_year <-   
 map(urls\_for\_each\_year\_headers,  
 ~str\_subset(tolower(.x),  
 paste0(job\_headings, collapse = "|")))  
  
total\_number\_of\_jobs\_per\_year\_n <-   
map\_int(total\_number\_of\_jobs\_per\_year, length)  
  
total\_number\_of\_jobs\_per\_year\_tbl <-   
tibble(  
 url\_to\_data\_source\_e\_g\_paste\_in\_url\_to\_the\_jobs\_wiki\_page = urls\_for\_each\_year,  
 total\_number\_of\_jobs\_per\_year = total\_number\_of\_jobs\_per\_year\_n  
)  
  
# count of TT jobs per year from our manual data collection,  
# join with our total number of all jobs by scraping  
count\_of\_tt\_jobs\_per\_year\_from\_our\_form <-   
jobdata %>%   
 group\_by(url\_to\_data\_source\_e\_g\_paste\_in\_url\_to\_the\_jobs\_wiki\_page) %>%   
 tally() %>%   
 right\_join(total\_number\_of\_jobs\_per\_year\_tbl) %>%   
 rename(n\_tt\_jobs = n,  
 n\_total\_jobs = total\_number\_of\_jobs\_per\_year) %>%   
 mutate(n\_non\_tt\_jobs = n\_total\_jobs - n\_tt\_jobs,  
 ratio\_tt\_2\_ntt = n\_tt\_jobs / n\_non\_tt\_jobs) %>%   
 mutate(year = str\_extract(url\_to\_data\_source\_e\_g\_paste\_in\_url\_to\_the\_jobs\_wiki\_page, "[[0-9]]{4}-[[0-9]]{4}|2021-22")) %>%   
 mutate(year = ifelse(year =="2021-22", "2021-2022", year))   
  
# draw plot  
fig\_ratio\_tt\_2\_ntt\_jobs\_per\_year <-   
 ggplot(count\_of\_tt\_jobs\_per\_year\_from\_our\_form) +  
 aes(year,   
 group = 1,  
 ratio\_tt\_2\_ntt) +  
 geom\_line(linewidth = 2) +  
 geom\_hline(yintercept = 1,  
 colour = "red") +  
 annotate("text",   
 x = 3,   
 y = 1.3,   
 label = "1:1 ratio",  
 colour = "red") +  
 labs(y = "Ratio of tenure-track\nto non-tenure track and other",  
 x = "") +  
 theme\_minimal(base\_size = 14) +  
 scale\_x\_discrete(name = "Academic Year")   
 #guides(x = "none")

# save these three plots as one set  
library(cowplot)  
plot\_grid(  
 fig\_how\_many\_jobs\_per\_year,  
 fig\_prop\_by\_job\_title\_per\_year,  
 fig\_ratio\_tt\_2\_ntt\_jobs\_per\_year,  
  
 ncol = 1,  
 align = "hv",  
 axis = "lr",  
 labels = "AUTO"  
)



ggsave(here("analysis",  
 "figures",   
 "fig-panel-per-year.png"),  
 bg ="white",  
 h = 11, # experiment with h and w to get the right size and proportion   
 w = 20,  
 units = "in",  
 dpi = 900) # make the image nice and crisp

# look at only these requirements because the others are flat  
  
intresting\_requirements <-   
c("cover letter",  
 "cv",  
 "names of recommenders",  
 "diversity statement",  
 "research statement",  
 "teaching statement")  
  
jobdata\_requirements <-   
jobdata %>%   
 select(year\_ad\_posted,  
 starts\_with("documents\_requested")) %>%   
 pivot\_longer(-year\_ad\_posted) %>%   
 mutate(value = case\_when(  
 value == "Not requested in the job ad" ~ 0,  
 value == "One" ~ 1,  
 value == "Two (e.g. two syllabi)" ~ 2,  
 value == "Three" ~ 3,  
 .default = 0  
 )) %>%   
 # trim names a bit  
 mutate(name = str\_remove(name, "documents\_requested\_")) %>%   
 mutate(name = str\_replace\_all(name, "\_", " ")) %>%   
 filter(name %in% intresting\_requirements) %>%   
 mutate(name = str\_wrap(name, 10),  
 year\_ad\_posted = str\_replace(year\_ad\_posted, "-", "\n"))  
  
jobdata\_requirements\_means <-   
jobdata\_requirements %>% # average number requested per year  
 group\_by(year\_ad\_posted,   
 name) %>%   
 summarise(mean\_n = mean(value))  
  
integer\_breaks <- function(n = 5, ...) {  
 fxn <- function(x) {  
 breaks <- floor(pretty(x, n, ...))  
 names(breaks) <- attr(breaks, "labels")  
 breaks  
 }  
 return(fxn)  
}  
  
  
ggplot(jobdata\_requirements\_means) +  
 aes(year\_ad\_posted,   
 mean\_n,  
 group = name) +  
 geom\_smooth(linewidth = 2,  
 colour = "black") +  
 geom\_jitter(data = jobdata\_requirements,  
 aes(year\_ad\_posted,   
 value),  
 alpha = 0.1,  
 height = 0.2,  
 width = 0.1) +  
 facet\_wrap(~name,  
 scales = "free\_y",  
 nrow = 1) +  
 xlab("Year") +  
 ylab("Number requested in job ad") +  
 scale\_y\_continuous(breaks = integer\_breaks()) +  
 theme\_minimal(base\_size = 14) +  
 theme(axis.text.x = element\_text(size = 8),  
 strip.text = element\_text( size = 20))  
  
ggsave(here("analysis",  
 "figures",   
 "fig-requirements-per-year.png"),  
 bg ="white",  
 h = 10, # experiment with h and w to get the right size and proportion   
 w = 20,  
 units = "in",  
 dpi = 900) # make the image nice and crisp

|  |
| --- |
| Figure 1 |

# do the requirements differ for associate positions   
jobdata\_requirements\_by\_rank <-   
jobdata %>%   
 mutate(position\_title = case\_when(  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,   
 "associate") ~ "associate",  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,   
 "assistant") ~ "assistant",  
 str\_detect(title\_of\_position\_tenure\_track\_jobs\_only,   
 "full") ~ "full")) %>%   
 select(position\_title,  
 starts\_with("documents\_requested")) %>%   
 pivot\_longer(-position\_title) %>%   
 mutate(value = case\_when(  
 value == "Not requested in the job ad" ~ 0,  
 value == "One" ~ 1,  
 value == "Two (e.g. two syllabi)" ~ 2,  
 value == "Three" ~ 3,  
 .default = 0  
 )) %>%   
 filter(!is.na(position\_title))   
  
jobdata\_requirements\_by\_rank\_means <-   
 jobdata\_requirements\_by\_rank %>%   
 group\_by(position\_title,  
 name) %>%   
 summarise(mean = mean(value))  
  
ggplot() +  
 geom\_jitter(data = jobdata\_requirements\_by\_rank,  
 aes(position\_title,   
 value),  
 height = 0.05,  
 alpha = 0.1) +  
 geom\_point(data = jobdata\_requirements\_by\_rank\_means,  
 aes(position\_title,  
 mean),  
 size = 4,  
 colour = "red") +  
 facet\_wrap( ~ name,  
 scales = "free\_y") +  
 theme\_minimal()

|  |
| --- |
| Figure 2 |

# geographic focus by year  
  
library(googlesheets4)  
library(stringi)  
  
geographic\_foci <-  
read\_sheet("https://docs.google.com/spreadsheets/d/1AHq49pIyChcgJ7rawe6KMWkdIBXydCamvg8Jslob8Ec/edit#gid=0", sheet = "geography")  
  
geographic\_foci\_clean <-  
 map(  
 str\_split(geographic\_foci$`From the data`, ";"),  
 ~.x %>%  
 str\_squish() %>%  
 stri\_remove\_empty())  
  
jobdata\_geo <-  
 jobdata %>%  
 select(geographic\_focus\_of\_position)  
  
jobdata\_geo <-  
 # add one column for each geo region in our categories  
cbind(jobdata\_geo,  
 setNames( lapply(geographic\_foci$Category2, function(x) x=NA),  
 geographic\_foci$Category2) )  
  
for(i in 1:length(geographic\_foci$Category2)){  
  
 this\_location <- geographic\_foci$Category2[i]  
  
 # create the pattern to search for  
 x <- paste0(geographic\_foci\_clean[[i]], collapse = "|")  
  
 # do the search through all the job ads for that pattern  
 y <- str\_detect(jobdata\_geo$geographic\_focus\_of\_position,  
 x)  
  
 # assign back to our data frame in the appropriate location column  
 jobdata\_geo[, this\_location] <- y  
  
}  
  
# BM TODO: check for job ads that have a location, but we're not getting it  
  
united\_states\_regions <-  
str\_subset(geographic\_foci$Category2, "US")  
  
jobdata\_geo\_year <-  
jobdata %>%  
 bind\_cols(jobdata\_geo) %>%  
 select(year\_ad\_posted,  
 geographic\_foci$Category2) %>%  
 pivot\_longer(-year\_ad\_posted) %>%  
 drop\_na()  
  
# how many times each location mentioned?  
jobdata\_geo\_year %>%  
 group\_by(name) %>%  
 summarise(n = sum(value)) %>%  
 arrange(desc(n)) %>%  
 ggplot() +  
 aes(reorder(name, n),  
 n)+  
 geom\_col() +  
 xlab("") +  
 theme\_minimal() +  
 coord\_flip()  
  
# explore trends over time. put a point on the max year  
jobdata\_geo\_year\_tally <-  
jobdata\_geo\_year %>%  
 # exclude those with <20 ads  
 filter(!name %in% c("Canada & Arctic",  
 "Oceania",  
 "Southeast US",  
 "Southwest US",  
 "Western US",  
 "Midwest US",  
 "Northeastern US"  
 )) %>%  
 group\_by(year\_ad\_posted,  
 name) %>%  
 summarise(n = sum(value)) %>%  
 mutate(prop = n / sum(n))  
  
jobdata\_geo\_year\_tally\_max <-  
 jobdata\_geo\_year\_tally %>%  
 group\_by(  
 name ) %>%  
 filter(prop == max(prop))  
  
library(ggrepel)  
  
ggplot() +  
 geom\_smooth(data = jobdata\_geo\_year\_tally,  
 aes(year\_ad\_posted,  
 prop,  
 group = name,  
 colour = name),  
 size = 3,  
 se = FALSE   
 ) +  
 xlab("Year") +  
 ylab("Proportion of all ads") +  
 scale\_colour\_brewer(palette = "Dark2") +  
 guides(colour = guide\_legend("Geographic\nfocus",  
 label.position = "bottom")) +  
 theme\_minimal( base\_size = 14) +  
 theme(legend.position="bottom")   
  
ggsave(here("analysis",  
 "figures",   
 "fig-geo-focus-by-year.png"),  
 bg ="white",  
 h = 10, # experiment with h and w to get the right size and proportion   
 w = 12,  
 units = "in",  
 dpi = 900) # make the image nice and crisp)  
  
  
# what about within the US  
# how many times each location mentioned?  
jobdata\_geo\_year %>%  
 group\_by(name) %>%  
 summarise(n = sum(value)) %>%  
 arrange(desc(n)) %>%  
 filter(name %in% united\_states\_regions) %>%   
 ggplot() +  
 aes(reorder(name, n),  
 n)+  
 geom\_col() +  
 ylab("Number of ads") +  
 xlab("") +  
 theme\_minimal(base\_size = 24) +  
 coord\_flip()  
  
ggsave(here("analysis",  
 "figures",   
 "fig-geo-us-focus-by-year.png"),  
 bg ="white",  
 h = 5, # experiment with h and w to get the right size and proportion   
 w = 6,  
 units = "in",  
 dpi = 900) # make the image nice and crisp)

|  |
| --- |
| Figure 3 |

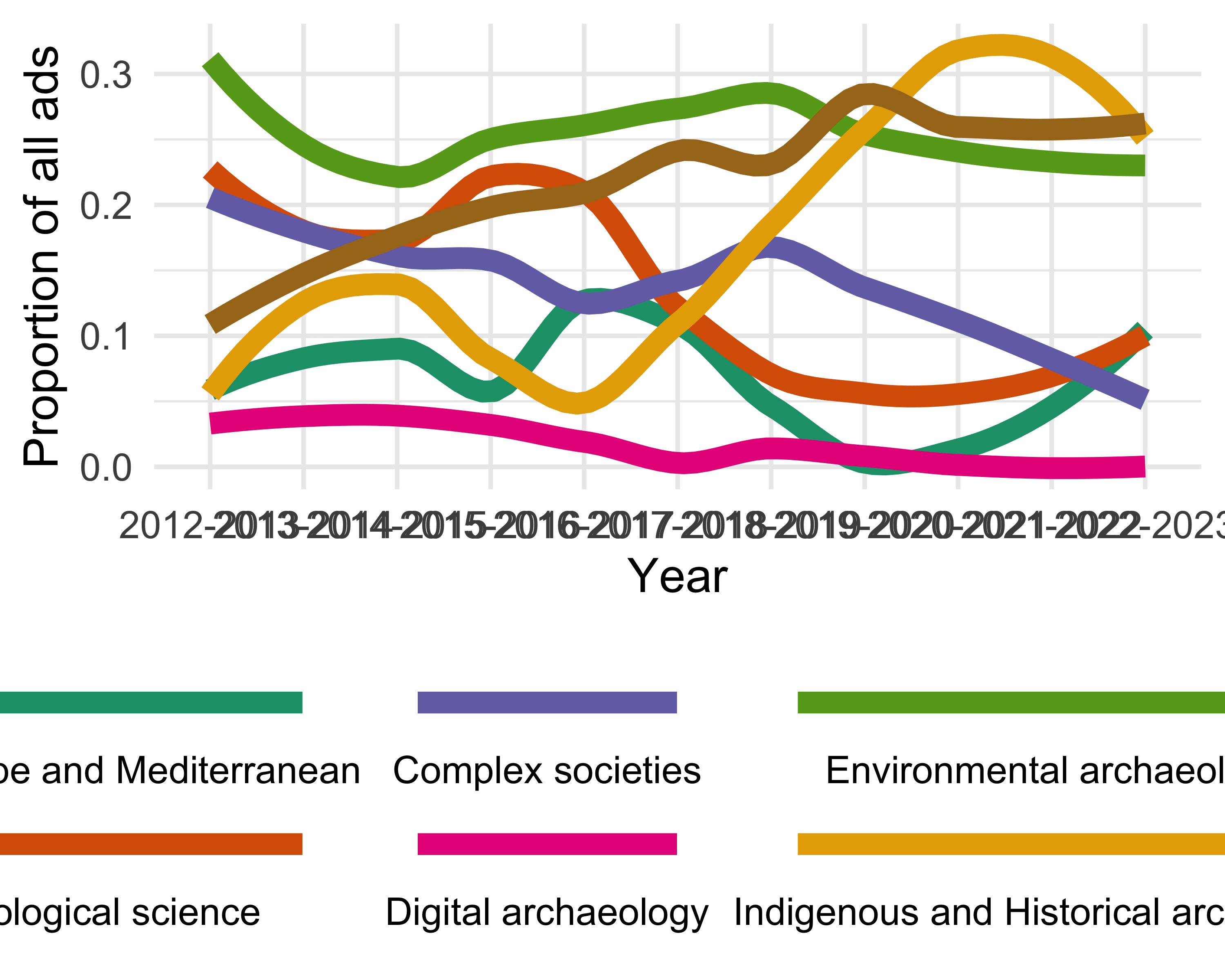
|  |
| --- |
| Figure 4 |

|  |
| --- |
| Figure 5 |

# topical focus by year  
  
library(googlesheets4)  
library(stringi)  
  
topical\_foci <-  
 read\_sheet("https://docs.google.com/spreadsheets/d/1AHq49pIyChcgJ7rawe6KMWkdIBXydCamvg8Jslob8Ec/edit#gid=0",  
 sheet = "topic")  
  
topical\_foci\_clean <-  
 map(  
 str\_split(topical\_foci$`From the data`, ";"),  
 ~.x %>%  
 str\_squish() %>%  
 stri\_remove\_empty() %>%  
 str\_to\_lower)  
  
jobdata\_topic <-  
 jobdata %>%  
 select(topical\_focus\_of\_position) %>%  
 mutate(topical\_focus\_of\_position = str\_to\_lower(topical\_focus\_of\_position))  
  
jobdata\_topic <-  
 # add one column for each topic in our categories  
 cbind(jobdata\_topic,  
 setNames( lapply(topical\_foci$Category, function(x) x=NA),  
 topical\_foci$Category) )  
  
for(i in 1:length(topical\_foci$Category)){  
  
 this\_topic <- topical\_foci$Category[i]  
  
 # create the pattern to search for  
 x <- paste0(topical\_foci\_clean[[i]], collapse = "|")  
  
 # do the search through all the job ads for that pattern  
 y <- str\_detect(jobdata\_topic$topical\_focus\_of\_position,  
 x)  
  
 # assign back to our data frame in the appropriate location column  
 jobdata\_topic[, this\_topic] <- y  
  
}  
  
jobdata\_topic\_year <-  
 jobdata %>%  
 bind\_cols(jobdata\_topic) %>%  
 select(year\_ad\_posted,  
 topical\_foci$Category) %>%  
 pivot\_longer(-year\_ad\_posted) %>%  
 drop\_na()  
  
# how many times each topic mentioned?  
jobdata\_topic\_year %>%  
 group\_by(name) %>%  
 summarise(n = sum(value)) %>%  
 arrange(desc(n)) %>%  
 ggplot() +  
 aes(reorder(name, n),  
 n)+  
 geom\_col() +  
 xlab("") +  
 theme\_minimal() +  
 coord\_flip()

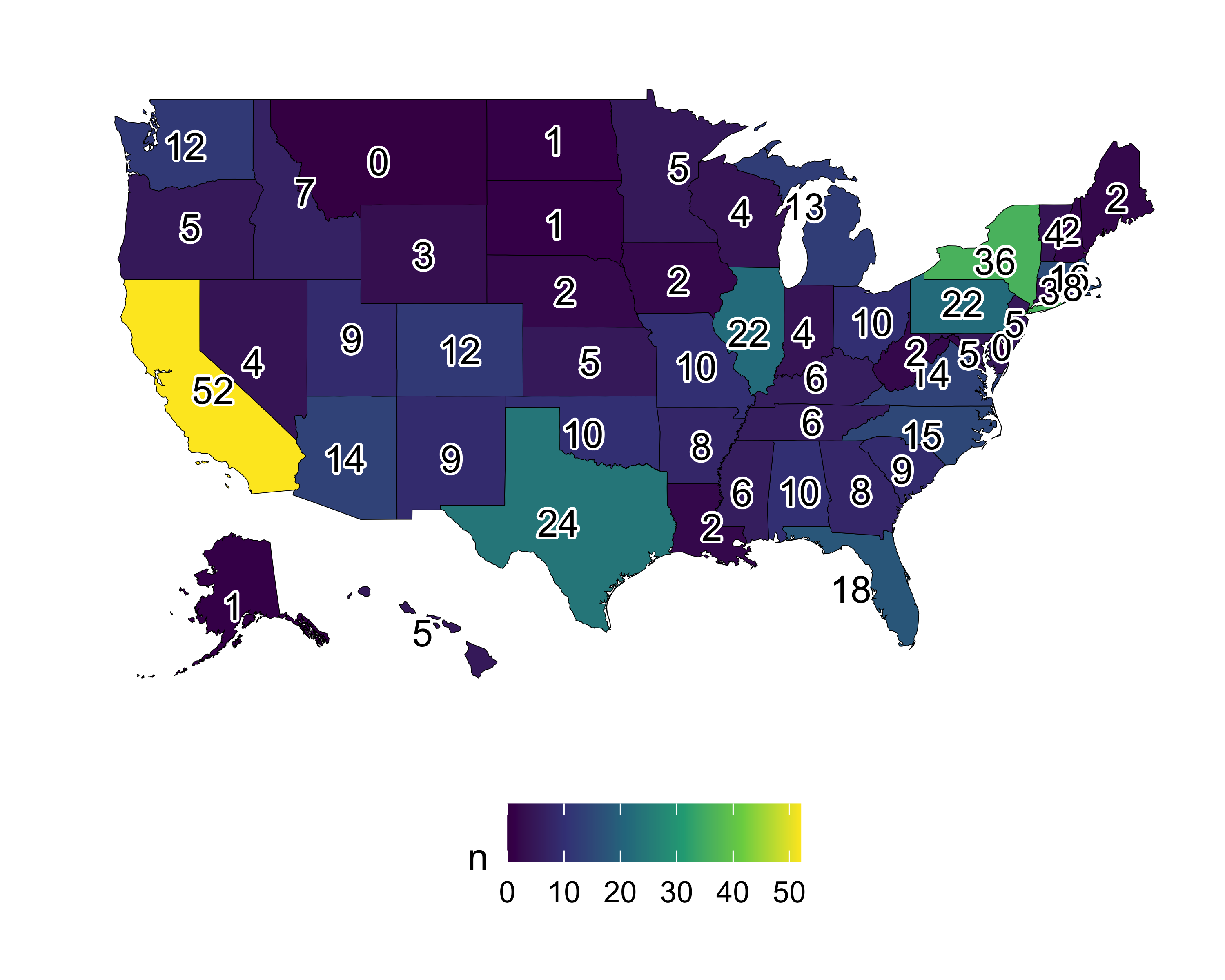


# explore trends over time. put a point on the max year  
jobdata\_topic\_year\_tally <-  
 jobdata\_topic\_year %>%  
 # exclude those with <20 ads  
 filter(!name %in% c("Digital Archaeology",  
 "Pleistocene archaeology",  
 "Mesoamerican Archaeology",  
 "Biological anthropology",  
 "Archaeological theory",  
 "Evolutionary anthropology",  
 "North American archaeology"  
 )) %>%  
 group\_by(year\_ad\_posted,  
 name) %>%  
 summarise(n = sum(value)) %>%  
 mutate(prop = n / sum(n))  
  
jobdata\_topic\_year\_tally\_max <-  
 jobdata\_topic\_year\_tally %>%  
 group\_by(  
 name ) %>%  
 filter(prop == max(prop))  
  
library(ggrepel)  
  
ggplot() +  
 geom\_smooth(data = jobdata\_topic\_year\_tally,  
 aes(year\_ad\_posted,  
 prop,  
 group = name,  
 colour = name),  
 size = 3,  
 span = 0.7,  
 se = FALSE) +  
 xlab("Year") +  
 ylab("Proportion of all ads") +  
 scale\_colour\_brewer(palette = "Dark2") +  
 guides(colour = guide\_legend("Topic\nfocus",  
 label.position = "bottom")) +  
 theme\_minimal( base\_size = 14) +  
 theme(legend.position="bottom")



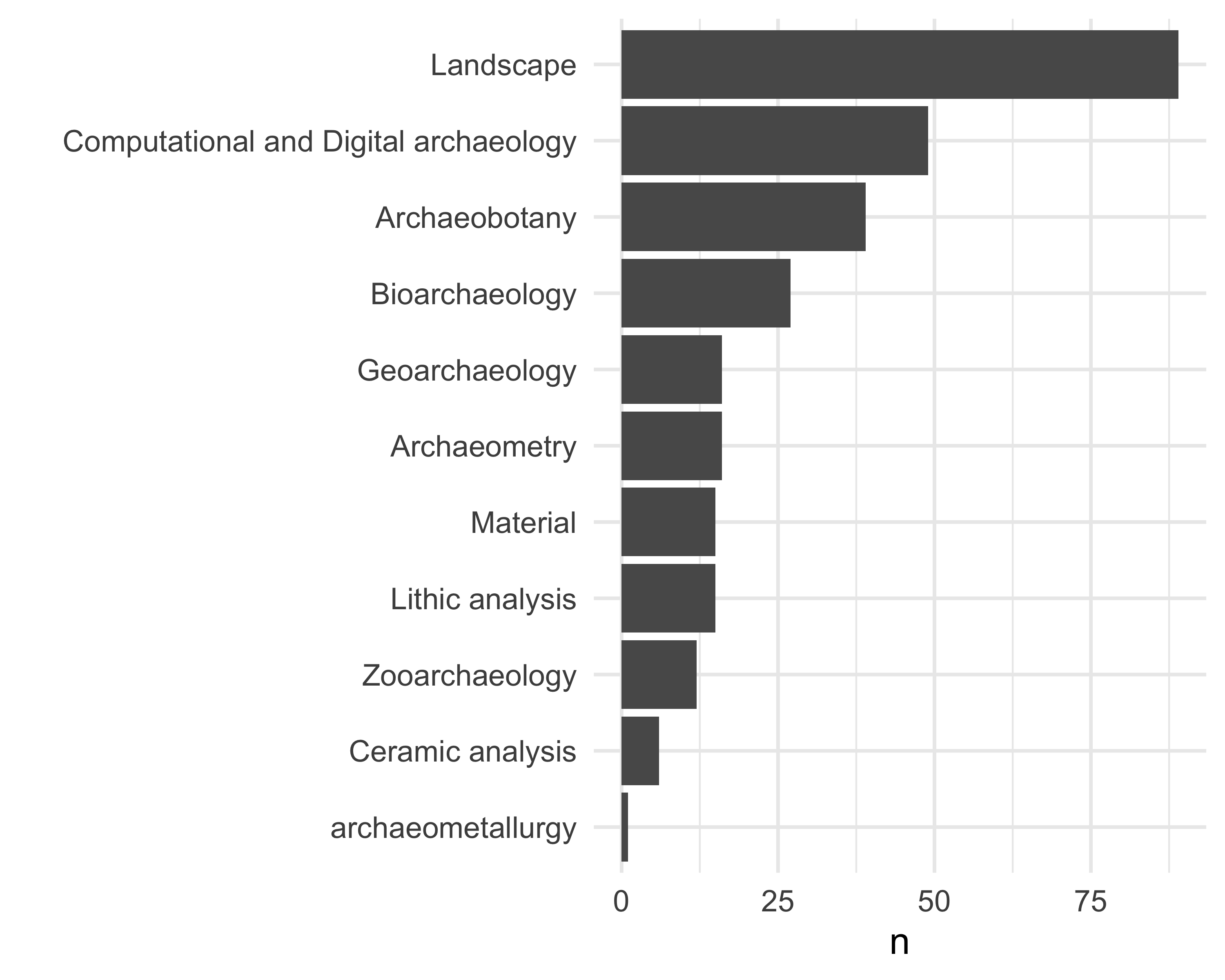
ggsave(here("analysis",  
 "figures",   
 "fig-topic-focus-by-year.png"),  
 bg ="white",  
 h = 10, # experiment with h and w to get the right size and proportion  
 w = 12,  
 units = "in",  
 dpi = 900) # make the image nice and crisp))

# Draw of map to show which states have done the most hiring in our sample  
  
# get the text in parentheses after the university name that gives the  
# state or country abb  
uni\_state\_country <- # 550  
jobdata %>% # 550 rows  
 select(name\_of\_hiring\_university) %>%  
 mutate(state\_country = regmatches(name\_of\_hiring\_university,  
 gregexpr( "(?<=\\().+?(?=\\))",  
 name\_of\_hiring\_university,  
 perl = T))) %>%  
 unnest(state\_country)  
  
# tally to get counts:  
uni\_state\_country\_tally <-  
uni\_state\_country %>%  
 group\_by(state\_country) %>%  
 tally(sort = TRUE)  
  
# did we get all the job ads?  
# sum(uni\_state\_country\_tally$n) # 550 all of them  
  
state\_to\_st <- function(x){  
 c(state.abb, 'DC')[match(x, c(state.name, 'District of Columbia'))]  
}  
  
state\_name\_and\_abb <-  
enframe(state.name, value = 'state\_name') %>%  
 mutate(state\_abbr = state\_to\_st(state\_name))  
  
# filter to get US states only  
uni\_state\_country\_tally\_us <-  
uni\_state\_country\_tally %>%  
 filter(state\_country %in% state.abb) %>%  
 select(state\_abbr = state\_country, n) %>%  
 # make sure we have all states in the dataframe  
 # even those with no jobs  
 right\_join(state\_name\_and\_abb) %>%  
 select(state = state\_name, n, state\_abbr) %>%  
 mutate(state = tolower(state)) %>%  
 mutate(n = ifelse(is.na(n), 0, n))  
  
# how many jobs ads now?  
# sum(uni\_state\_country\_tally\_us$n) # 433, 78% of the total  
  
library(ggplot2)  
library(fiftystater)  
library(tidyverse)  
library(ggrepel)  
  
data("fifty\_states")  
  
ggplot(data= uni\_state\_country\_tally\_us,  
 aes(map\_id = state)) +  
 geom\_map(aes(fill = n),  
 color= "black",  
 linewidth = 0.1,  
 map = fifty\_states) +  
 expand\_limits(x = fifty\_states$long,  
 y = fifty\_states$lat) +  
 coord\_map() +  
 geom\_text\_repel(data = fifty\_states %>%  
 group\_by(id) %>%  
 summarise(lat = mean(c(max(lat), min(lat))),  
 long = mean(c(max(long), min(long)))) %>%  
 mutate(state = id) %>%  
 left\_join(uni\_state\_country\_tally\_us,  
 by = "state"),  
 aes(x = long,  
 y = lat,  
 label = n,  
 bg.color = "white",  
 bg.r = 0.1),  
 force = 0,  
 force\_pull = 100)+  
 scale\_x\_continuous(breaks = NULL) +  
 scale\_y\_continuous(breaks = NULL) +  
 labs(x = "",  
 y = "") +  
 theme(legend.position = "bottom",  
 panel.background = element\_blank()) +  
 scale\_fill\_viridis\_c()

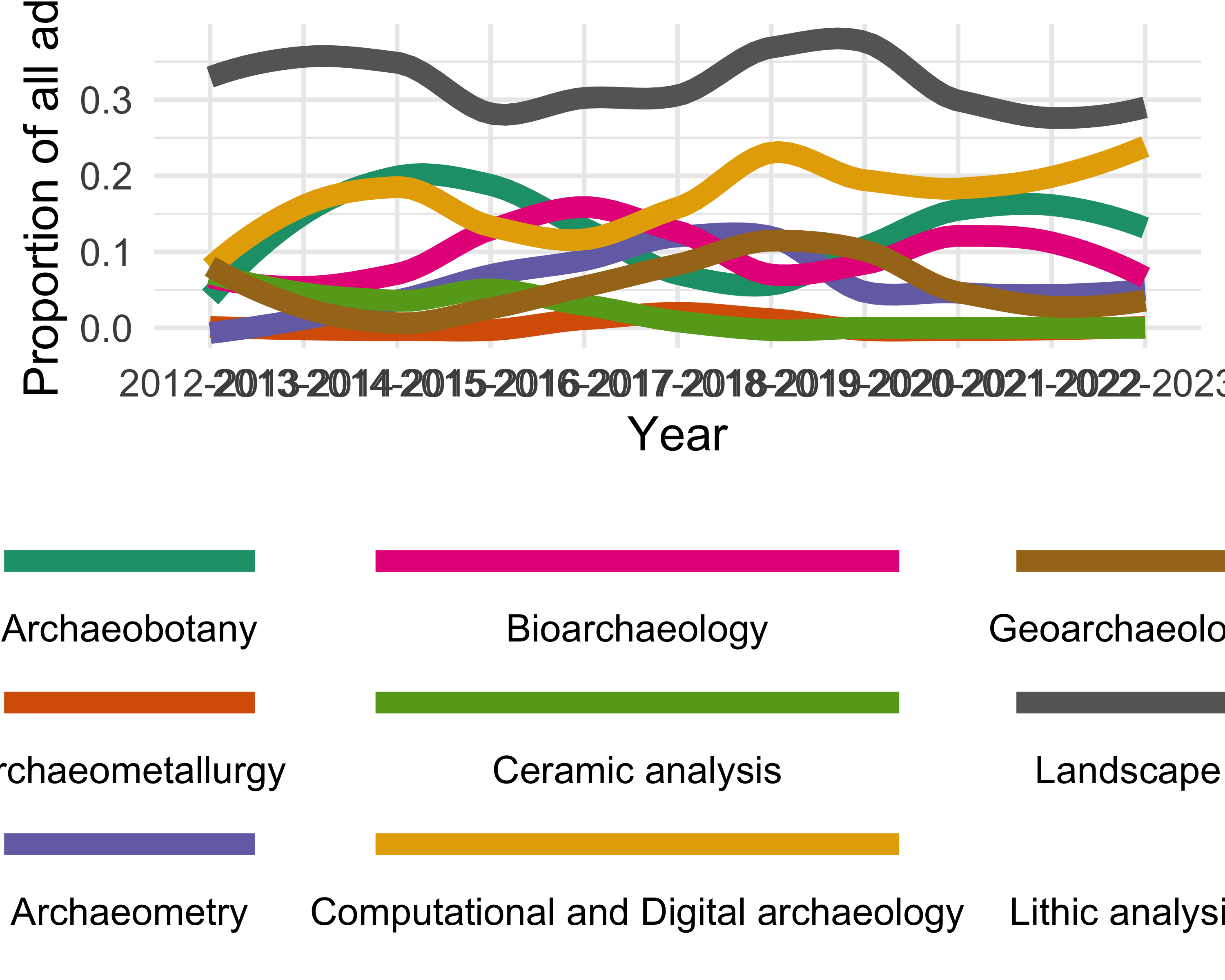


ggsave(here("analysis",  
 "figures",  
 "fig-us-state-map.png"),  
 bg ="white",  
 h = 10, # experiment with h and w to get the right size and proportion  
 w = 12,  
 units = "in",  
 dpi = 900) # make the image nice and crisp))

# method focus by year  
  
library(googlesheets4)  
library(stringi)  
  
method\_foci <-  
 read\_sheet("https://docs.google.com/spreadsheets/d/1AHq49pIyChcgJ7rawe6KMWkdIBXydCamvg8Jslob8Ec/edit#gid=0",  
 sheet = "method")  
  
method\_foci\_clean <-  
 map(  
 str\_split(method\_foci$`From the data`, ";"),  
 ~.x %>%  
 str\_squish() %>%  
 stri\_remove\_empty() %>%  
 str\_to\_lower)  
  
jobdata\_method <-  
 jobdata %>%  
 select(methods\_focus\_of\_position) %>%  
 mutate(methods\_focus\_of\_position = str\_to\_lower(methods\_focus\_of\_position))  
  
jobdata\_method <-  
 # add one column for each topic in our categories  
 cbind(jobdata\_method,  
 setNames( lapply(method\_foci$Category, function(x) x=NA),  
 method\_foci$Category) )  
  
for(i in 1:length(method\_foci$Category)){  
  
 this\_method <- method\_foci$Category[i]  
  
 # create the pattern to search for  
 x <- paste0(method\_foci\_clean[[i]], collapse = "|")  
  
 # do the search through all the job ads for that pattern  
 y <- str\_detect(jobdata\_method$methods\_focus\_of\_position,  
 x)  
  
 # assign back to our data frame in the appropriate location column  
 jobdata\_method[, this\_method] <- y  
  
}  
  
jobdata\_method\_year <-  
 jobdata %>%  
 bind\_cols(jobdata\_method) %>%  
 select(year\_ad\_posted,  
 method\_foci$Category) %>%  
 pivot\_longer(-year\_ad\_posted) %>%  
 drop\_na()  
  
# how many times each topic mentioned?  
jobdata\_method\_year %>%  
 group\_by(name) %>%  
 summarise(n = sum(value)) %>%  
 arrange(desc(n)) %>%  
 ggplot() +  
 aes(reorder(name, n),  
 n)+  
 geom\_col() +  
 xlab("") +  
 theme\_minimal() +  
 coord\_flip()



# explore trends over time. put a point on the max year  
jobdata\_method\_year\_tally <-  
 jobdata\_method\_year %>%  
 # exclude those with <20 ads  
 filter(!name %in% c("Digital Archaeology",  
 "Pleistocene archaeology",  
 "Mesoamerican Archaeology",  
 "Biological anthropology",  
 "Archaeological theory",  
 "Evolutionary anthropology",  
 "North American archaeology"  
 )) %>%  
 group\_by(year\_ad\_posted,  
 name) %>%  
 summarise(n = sum(value)) %>%  
 mutate(prop = n / sum(n))  
  
jobdata\_method\_year\_tally\_max <-  
 jobdata\_method\_year\_tally %>%  
 group\_by(  
 name ) %>%  
 filter(prop == max(prop))  
  
library(ggrepel)  
  
ggplot() +  
 geom\_smooth(data = jobdata\_method\_year\_tally,  
 aes(year\_ad\_posted,  
 prop,  
 group = name,  
 colour = name),  
 size = 3,  
 span = 0.7,  
 se = FALSE) +  
 xlab("Year") +  
 ylab("Proportion of all ads") +  
 scale\_colour\_brewer(palette = "Dark2") +  
 guides(colour = guide\_legend("Method\nfocus",  
 label.position = "bottom")) +  
 theme\_minimal( base\_size = 14) +  
 theme(legend.position="bottom")



ggsave(here("analysis",  
 "figures",   
 "fig-method-focus-by-year.png"),  
 bg ="white",  
 h = 10, # experiment with h and w to get the right size and proportion  
 w = 12,  
 units = "in",  
 dpi = 900) # make the image nice and crisp))

# Discussion

# Conclusion

# Acknowledgements

# References

Marwick, B., 2017. Computational reproducibility in archaeological research: Basic principles and a case study of their implementation. Journal of Archaeological Method and Theory 24, 424–450. <https://doi.org/10.1007/s10816-015-9272-9>

### Colophon

This report was generated on 2024-04-17 15:36:42.558665 using the following computational environment and dependencies:

# which R packages and versions?  
if ("devtools" %in% installed.packages()) devtools::session\_info()

─ Session info ───────────────────────────────────────────────────────────────  
 setting value  
 version R version 4.3.3 (2024-02-29)  
 os macOS Sonoma 14.4.1  
 system x86\_64, darwin20  
 ui X11  
 language (EN)  
 collate en\_US.UTF-8  
 ctype en\_US.UTF-8  
 tz America/Los\_Angeles  
 date 2024-04-17  
 pandoc 3.1.1 @ /Applications/RStudio.app/Contents/Resources/app/quarto/bin/tools/ (via rmarkdown)  
  
─ Packages ───────────────────────────────────────────────────────────────────  
 ! package \* version date (UTC) lib source  
 P askpass 1.2.0 2023-09-03 [?] CRAN (R 4.3.0)  
 P beeswarm 0.4.0 2021-06-01 [?] CRAN (R 4.3.0)  
 P bit 4.0.5 2022-11-15 [?] CRAN (R 4.3.0)  
 P bit64 4.0.5 2020-08-30 [?] CRAN (R 4.3.0)  
 P cachem 1.0.8 2023-05-01 [?] CRAN (R 4.3.0)  
 P cellranger 1.1.0 2016-07-27 [?] CRAN (R 4.3.0)  
 P cli 3.6.2 2023-12-11 [?] CRAN (R 4.3.0)  
 P colorspace 2.1-0 2023-01-23 [?] CRAN (R 4.3.0)  
 P cowplot \* 1.1.3 2024-01-22 [?] CRAN (R 4.3.2)  
 P crayon 1.5.2 2022-09-29 [?] CRAN (R 4.3.0)  
 curl 5.2.0 2023-12-08 [1] RSPM (R 4.3.0)  
 P devtools 2.4.5 2022-10-11 [?] RSPM (R 4.3.0)  
 digest 0.6.34 2024-01-11 [1] RSPM (R 4.3.0)  
 P dplyr \* 1.1.4 2023-11-17 [?] CRAN (R 4.3.0)  
 P ellipsis 0.3.2 2021-04-29 [?] CRAN (R 4.3.0)  
 P evaluate 0.23 2023-11-01 [?] CRAN (R 4.3.0)  
 fansi 1.0.6 2023-12-08 [1] RSPM (R 4.3.0)  
 P farver 2.1.1 2022-07-06 [?] CRAN (R 4.3.0)  
 P fastmap 1.1.1 2023-02-24 [?] CRAN (R 4.3.0)  
 fiftystater \* 1.0.1 2024-04-06 [1] Github (wmurphyrd/fiftystater@28e7fa5)  
 P forcats \* 1.0.0 2023-01-29 [?] CRAN (R 4.3.0)  
 P fs 1.6.3 2023-07-20 [?] CRAN (R 4.3.0)  
 P gargle 1.5.2 2023-07-20 [?] CRAN (R 4.3.0)  
 P generics 0.1.3 2022-07-05 [?] CRAN (R 4.3.0)  
 P ggbeeswarm \* 0.7.2 2023-04-29 [?] CRAN (R 4.3.0)  
 P ggplot2 \* 3.4.4 2023-10-12 [?] CRAN (R 4.3.0)  
 P ggrepel \* 0.9.5 2024-01-10 [?] CRAN (R 4.3.0)  
 P glue 1.7.0 2024-01-09 [?] CRAN (R 4.3.0)  
 P googledrive 2.1.1 2023-06-11 [?] CRAN (R 4.3.0)  
 P googlesheets4 \* 1.1.1 2023-06-11 [?] CRAN (R 4.3.0)  
 P gtable 0.3.4 2023-08-21 [?] CRAN (R 4.3.0)  
 P here \* 1.0.1 2020-12-13 [?] CRAN (R 4.3.0)  
 P hms 1.1.3 2023-03-21 [?] CRAN (R 4.3.0)  
 P htmltools 0.5.7 2023-11-03 [?] CRAN (R 4.3.0)  
 htmlwidgets 1.6.4 2023-12-06 [1] RSPM (R 4.3.0)  
 httpuv 1.6.14 2024-01-26 [1] RSPM (R 4.3.0)  
 P httr 1.4.7 2023-08-15 [?] CRAN (R 4.3.0)  
 P janitor 2.2.0 2023-02-02 [?] CRAN (R 4.3.0)  
 jsonlite 1.8.8 2023-12-04 [1] RSPM (R 4.3.0)  
 P knitr 1.45 2023-10-30 [?] CRAN (R 4.3.0)  
 P labeling 0.4.3 2023-08-29 [?] CRAN (R 4.3.0)  
 later 1.3.2 2023-12-06 [1] RSPM (R 4.3.0)  
 lattice 0.22-5 2023-10-24 [2] CRAN (R 4.3.3)  
 P lifecycle 1.0.4 2023-11-07 [?] CRAN (R 4.3.0)  
 P lubridate \* 1.9.3 2023-09-27 [?] CRAN (R 4.3.0)  
 P magrittr 2.0.3 2022-03-30 [?] CRAN (R 4.3.0)  
 mapproj 1.2.11 2023-01-12 [1] CRAN (R 4.3.0)  
 P maps 3.4.2 2023-12-15 [?] CRAN (R 4.3.0)  
 Matrix 1.6-5 2024-01-11 [2] CRAN (R 4.3.3)  
 P memoise 2.0.1 2021-11-26 [?] CRAN (R 4.3.0)  
 mgcv 1.9-1 2023-12-21 [2] CRAN (R 4.3.3)  
 P mime 0.12 2021-09-28 [?] CRAN (R 4.3.0)  
 P miniUI 0.1.1.1 2018-05-18 [?] RSPM (R 4.3.0)  
 P munsell 0.5.0 2018-06-12 [?] CRAN (R 4.3.0)  
 nlme 3.1-164 2023-11-27 [2] CRAN (R 4.3.3)  
 P openssl 2.1.1 2023-09-25 [?] CRAN (R 4.3.0)  
 P pillar 1.9.0 2023-03-22 [?] CRAN (R 4.3.0)  
 pkgbuild 1.4.3 2023-12-10 [1] RSPM (R 4.3.0)  
 P pkgconfig 2.0.3 2019-09-22 [?] CRAN (R 4.3.0)  
 pkgload 1.3.4 2024-01-16 [1] RSPM (R 4.3.0)  
 P profvis 0.3.8 2023-05-02 [?] RSPM (R 4.3.0)  
 P promises 1.2.1 2023-08-10 [?] CRAN (R 4.3.0)  
 P purrr \* 1.0.2 2023-08-10 [?] CRAN (R 4.3.0)  
 P R6 2.5.1 2021-08-19 [?] CRAN (R 4.3.0)  
 ragg 1.2.7 2023-12-11 [1] RSPM (R 4.3.0)  
 P rappdirs 0.3.3 2021-01-31 [?] CRAN (R 4.3.0)  
 P RColorBrewer 1.1-3 2022-04-03 [?] CRAN (R 4.3.0)  
 Rcpp 1.0.12 2024-01-09 [1] RSPM (R 4.3.0)  
 P readr \* 2.1.5 2024-01-10 [?] CRAN (R 4.3.0)  
 P remotes 2.4.2.1 2023-07-18 [?] RSPM (R 4.3.0)  
 P rlang 1.1.3 2024-01-10 [?] CRAN (R 4.3.0)  
 P rmarkdown 2.25 2023-09-18 [?] CRAN (R 4.3.0)  
 rprojroot 2.0.4 2023-11-05 [1] RSPM (R 4.3.0)  
 P rstudioapi 0.15.0 2023-07-07 [?] CRAN (R 4.3.0)  
 P rvest \* 1.0.3 2022-08-19 [?] CRAN (R 4.3.0)  
 P scales 1.3.0 2023-11-28 [?] CRAN (R 4.3.0)  
 P selectr 0.4-2 2019-11-20 [?] CRAN (R 4.3.0)  
 P sessioninfo 1.2.2 2021-12-06 [?] RSPM (R 4.3.0)  
 P shiny 1.8.0 2023-11-17 [?] CRAN (R 4.3.0)  
 P snakecase 0.11.1 2023-08-27 [?] CRAN (R 4.3.0)  
 stringi \* 1.8.3 2023-12-11 [1] RSPM (R 4.3.0)  
 P stringr \* 1.5.1 2023-11-14 [?] CRAN (R 4.3.0)  
 P systemfonts 1.0.5 2023-10-09 [?] CRAN (R 4.3.0)  
 textshaping 0.3.7 2023-10-09 [1] RSPM (R 4.3.0)  
 P tibble \* 3.2.1 2023-03-20 [?] CRAN (R 4.3.0)  
 P tidyr \* 1.3.1 2024-01-24 [?] CRAN (R 4.3.2)  
 P tidyselect 1.2.0 2022-10-10 [?] CRAN (R 4.3.0)  
 P tidyverse \* 2.0.0 2023-02-22 [?] CRAN (R 4.3.0)  
 P timechange 0.3.0 2024-01-18 [?] CRAN (R 4.3.0)  
 P tzdb 0.4.0 2023-05-12 [?] CRAN (R 4.3.0)  
 P urlchecker 1.0.1 2021-11-30 [?] RSPM (R 4.3.0)  
 P usethis 2.2.2 2023-07-06 [?] RSPM (R 4.3.0)  
 P utf8 1.2.4 2023-10-22 [?] CRAN (R 4.3.0)  
 vctrs 0.6.5 2023-12-01 [1] RSPM (R 4.3.0)  
 P vipor 0.4.7 2023-12-18 [?] CRAN (R 4.3.0)  
 P viridisLite 0.4.2 2023-05-02 [?] CRAN (R 4.3.0)  
 P vroom 1.6.5 2023-12-05 [?] CRAN (R 4.3.0)  
 withr 3.0.0 2024-01-16 [1] RSPM (R 4.3.0)  
 P xfun 0.41 2023-11-01 [?] CRAN (R 4.3.0)  
 xml2 1.3.6 2023-12-04 [1] RSPM (R 4.3.0)  
 P xtable 1.8-4 2019-04-21 [?] RSPM (R 4.3.0)  
 yaml 2.3.8 2023-12-11 [1] RSPM (R 4.3.0)  
  
 [1] /Users/bmarwick/Library/Caches/org.R-project.R/R/renv/library/archyjobads-bfdb6333/R-4.3/x86\_64-apple-darwin20  
 [2] /Library/Frameworks/R.framework/Versions/4.3-x86\_64/Resources/library  
  
 P ── Loaded and on-disk path mismatch.  
  
──────────────────────────────────────────────────────────────────────────────

The current Git commit details are:

# what commit is this file at?   
if ("git2r" %in% installed.packages() & git2r::in\_repository(path = ".")) git2r::repository(here::here())

Local: main /Users/bmarwick/Downloads/archyjobads  
Remote: main @ origin (https://github.com/benmarwick/archyjobads)  
Head: [10e1fe2] 2024-04-17: merge conflict