MxNet Contributor Email Summary

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Overview

Email addresses are not available through the Github API and most profiles do not have a Company listed. Not all profiles provided names or the names provided were incomplete or "fake".

The hypotheses explored here are as follows: * Additional names can be extracted from a user's commit history * Email addresses can be extracted from a user's commit history * Company can be identified from the email domain name

The null hypotheses are as follows:

- Users don't have sufficient commit history to yield email addresses or names
- The majority of email addresses are the default obfuscated Github ones
- Users don't use their company email, therefore company cannot be derived from the email domain.

```
actors <- read.csv("mxnet_top_actors.csv", na.strings="")

total_actors = nrow(actors)

actors <- actors %>% mutate(
  commits_emails_cnt = commits_found_in_fork_cnt + commits_found_in_repo_cnt,
  has_name = !is.na(name),
  has_company = !is.na(company),
  updated_days = as.numeric(round(difftime(Sys.time(), updated_at))),
  updated_days_log = round(log(updated_days)),
  age_years = as.numeric(round(difftime(Sys.time(), created_at)/365)),
  has_commit_name = !is.na(commits_names),
  has_commit_email = commits_emails_cnt > 0
)

actors_updated_summary <- actors %>% group_by(updated_days_log) %>%
  summarise(
    updated_days_min = min(updated_days),
    updated_days_max = max(updated_days)
  ) %>%
  mutate(updated_min_max = ifelse(is.na(updated_days_min), NA,
                                  paste(updated_days_min, "-", updated_days_max)))

actors <- merge(actors, actors_updated_summary, by="updated_days_log")

rm(actors_updated_summary)
```

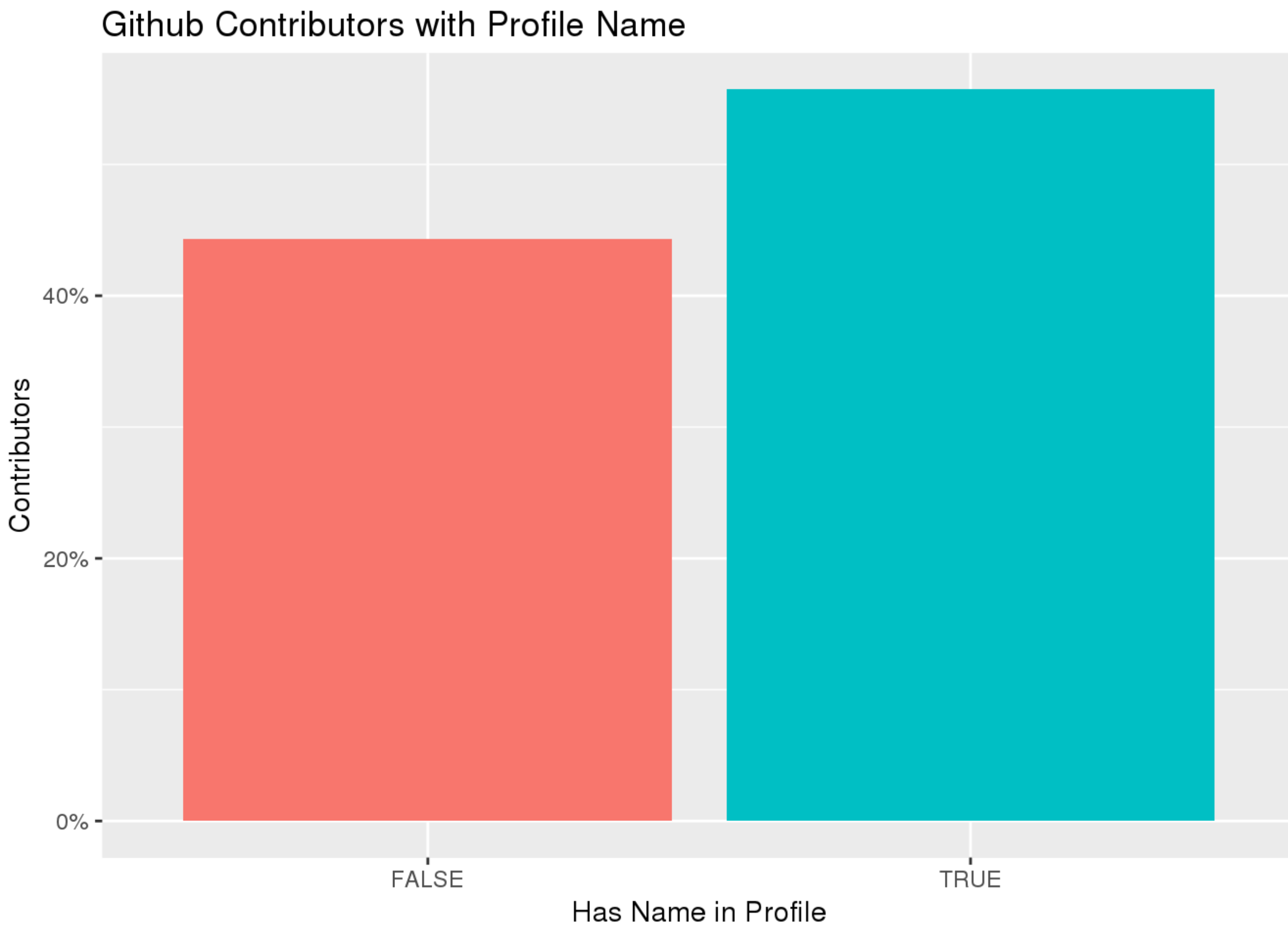
Profile Analysis

This section looks at what data are available from the GitHub profile. About 50% of the contributors have information in their profile such as name or company. Email addresses are not available from the Github API.

Name

Just under 60% of the contributors had provided a name in their Github profile. Profiles with names skew towards being older with the majority being about 2-4 years old.

```
ggplot(data = actors %>% group_by(has_name) %>% summarise(has_name_pct = n()/total_act
ors),
       aes(x=has_name, y=has_name_pct, fill=has_name)) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Has Name in Profile") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide=FALSE) +
  labs(title="Github Contributors with Profile Name")
```




```
ggsave("mxnet_profile_name.png")
```

```
## Saving 7 x 5 in image
```

```
ggplot(data = actors,
       aes(x=has_name, fill=factor(age_years))) +
  geom_bar(position="dodge") +
  xlab("Has Name in Profile") +
  ylab("Contributors") +
  scale_fill_discrete("Profile Age (years)") +
  labs(title="Github Contributors with Profile Name")
```


```
ggsave("mxnet_profile_name_age.png")
```

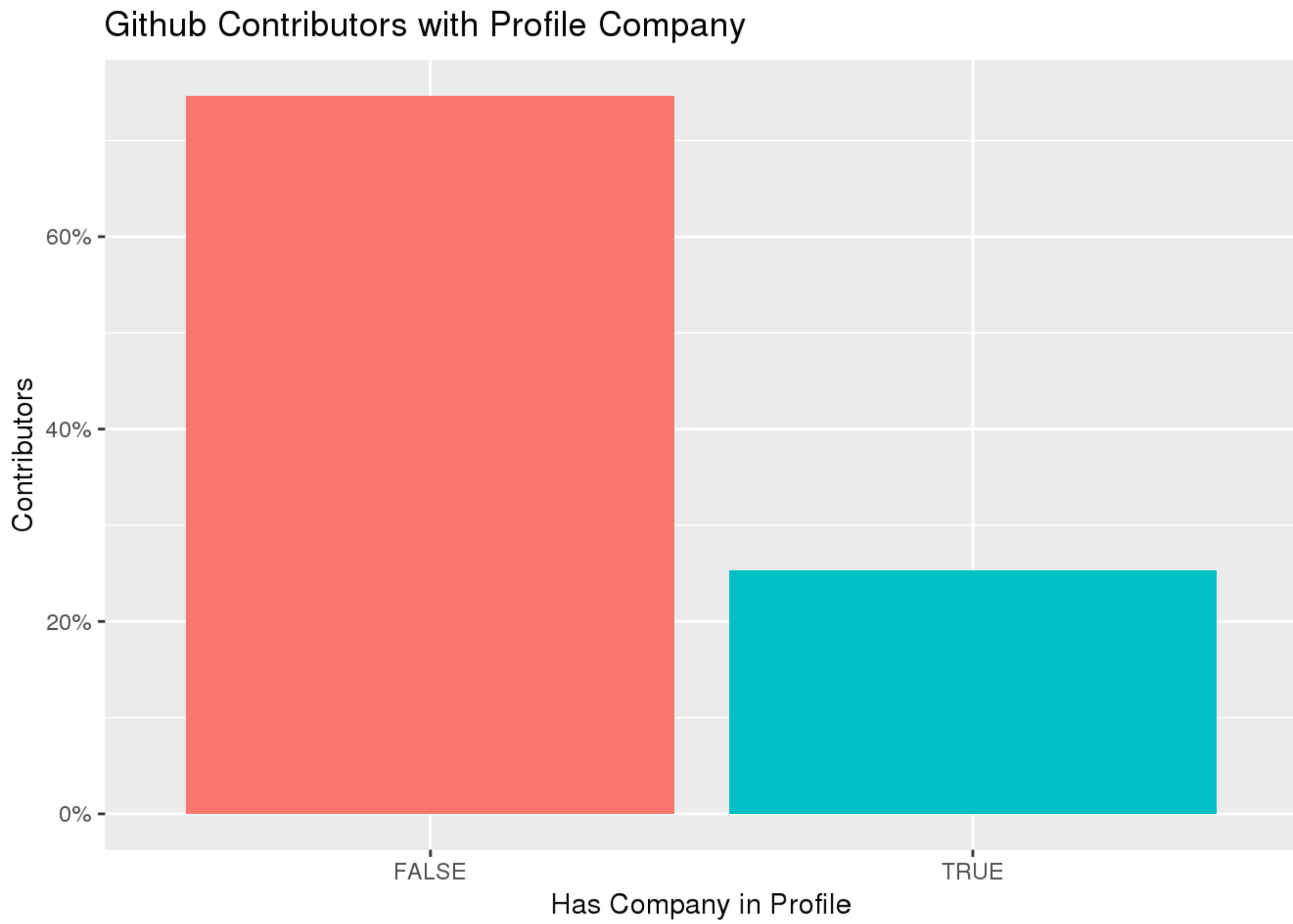
```
## Saving 7 x 5 in image
```

Company

Around 75% of the profiles do not have company information. For ones that do, how recent is this information? Are the companies reported largely accurate?

The majority of profiles, regardless of whether they had a company or not, were updated within 3 months. Profiles that were updated less recently did not tend to have company information. This suggests the company information should be up to date.

```
ggplot(data = actors %>% group_by(has_company) %>% summarise(has_company_pct = n()/tot
al_actors),
       aes(x=has_company, y=has_company_pct, fill=has_company)) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Has Company in Profile") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide=FALSE) +
  labs(title="Github Contributors with Profile Company")
```




```
ggsave("mxnet_profile_company.png")
```

```
## Saving 7 x 5 in image
```

```
ggplot(data = actors %>% group_by(has_company, updated_days_log) %>%
         summarise(has_company_pct = n()/total_actors, updated_min_max = first(updated
_min_max)),
       aes(x=has_company, y=has_company_pct, fill=reorder(updated_min_max, updated_day
s_log))) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Has Company in Profile") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete("Last Updated (days)") +
  labs(title="Github Contributors with Profile Company")
```


```
ggsave("mxnet_profile_company_updated.png")
```

```
## Saving 7 x 5 in image
```

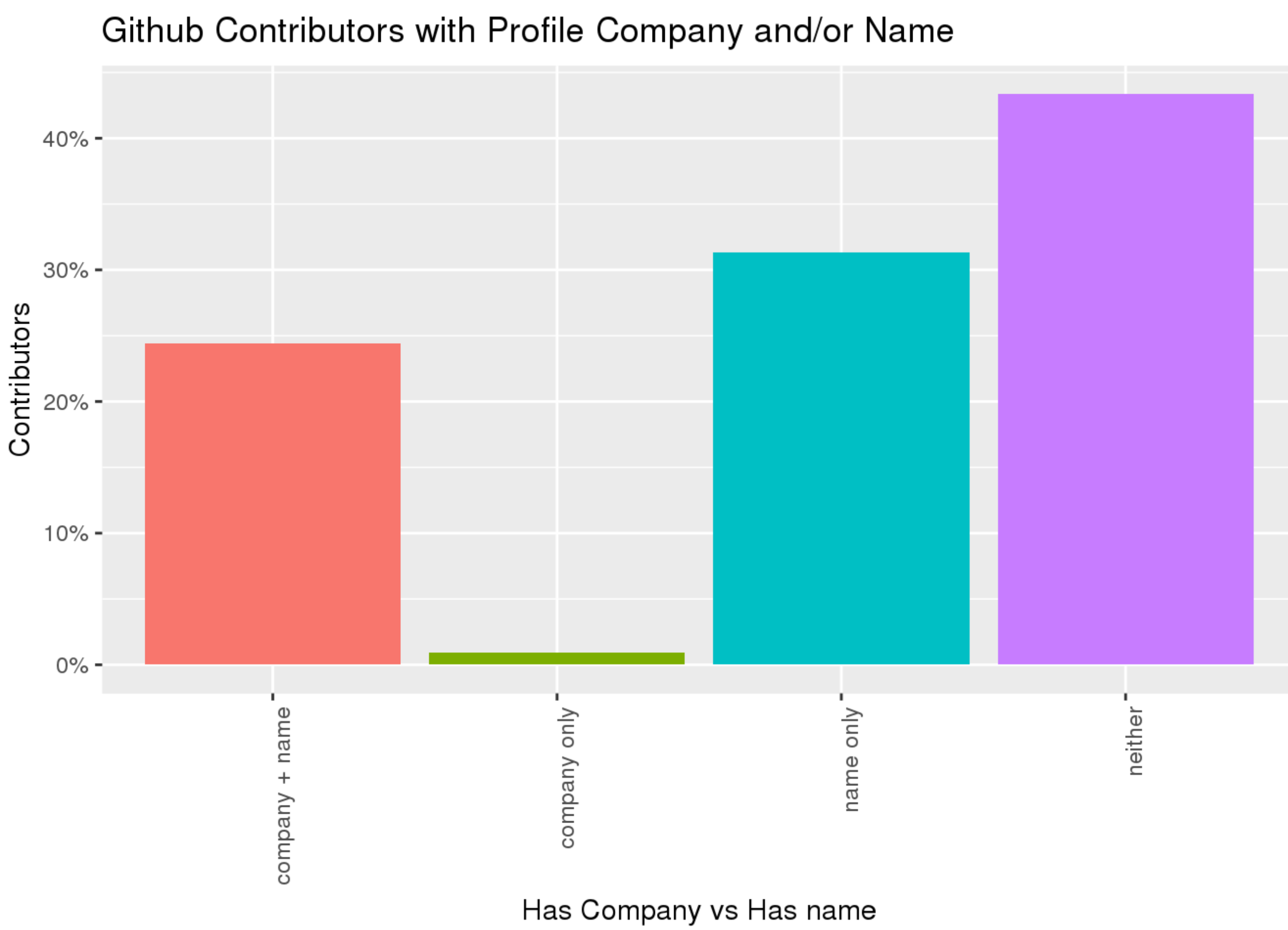
Name vs Company

Another interesting question is whether profiles with a company name also have a name. For ones without a name, is the company information potentially less accurate?

Just over 55% of the profiles had either a company or a name. 30% had a name only and 25% had both a company and a name. A very small percent had company only. Because these appear in such a small proportion, we should look at the value of the field for those.

We've already established that most of the Github profiles are fairly up to date, however it's worth looking at that distribution in terms of company vs name. We see a fairly similar distribution suggesting that the profiles are fairly up to date and neither parameter, name nor company, skews either way.

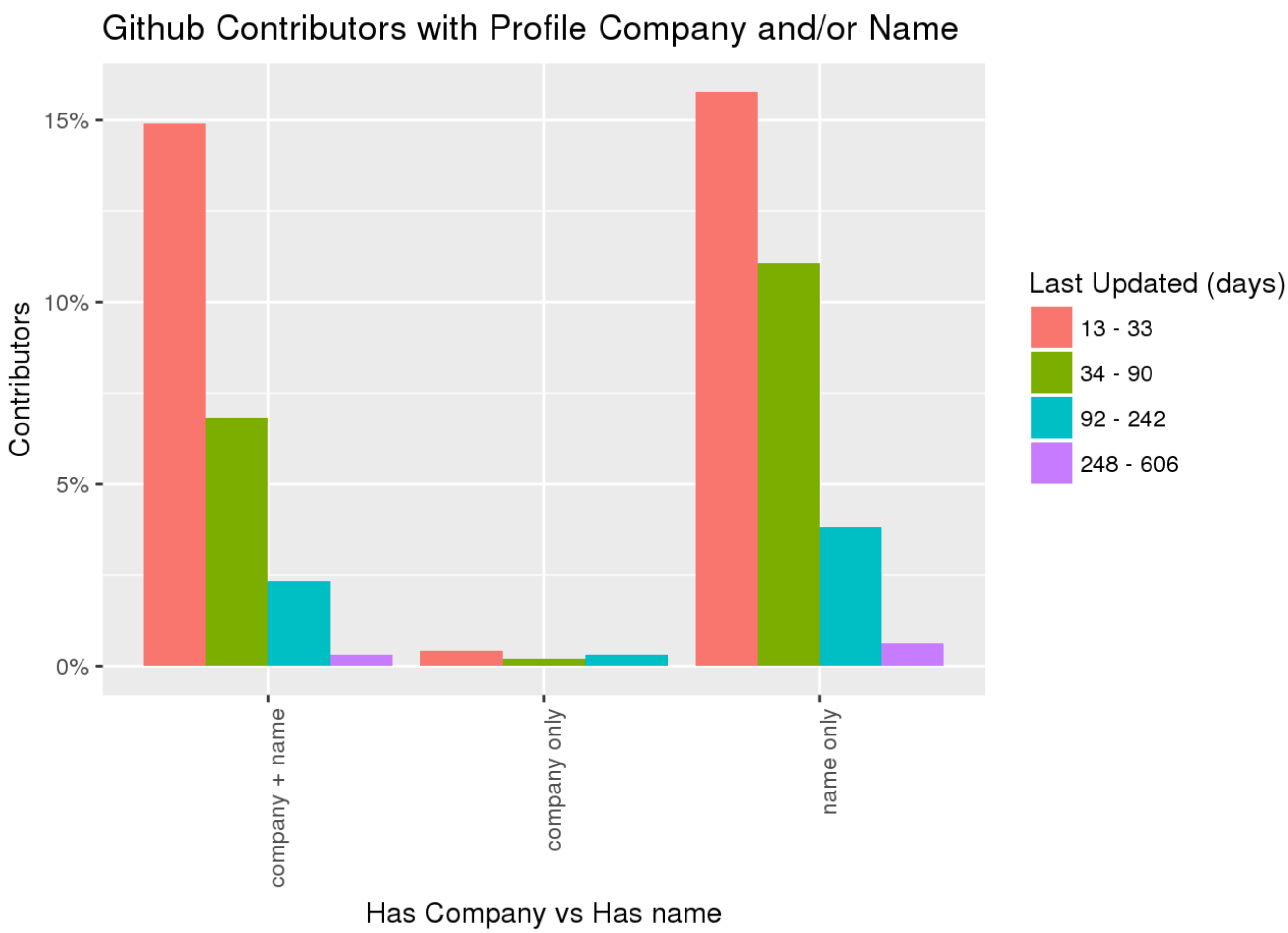
```r
# has company vs has name

actors <- actors %>%
  mutate(company_vs_name = ifelse(has_company & has_name, paste("company + name"),
                                  ifelse(has_company, paste("company only"),
                                  ifelse(has_name, paste("name only"),
                                         paste("neither")
                                  )))
  )

ggplot(data = actors %>% group_by(company_vs_name) %>% summarise(company_vs_name_pct =
 n()/total_actors),
       aes(x=company_vs_name, y=company_vs_name_pct, fill=company_vs_name)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Has Company vs Has name") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide=FALSE) +
  labs(title="Github Contributors with Profile Company and/or Name")
```




```r
ggsave("company_vs_name.png")
```

```
## Saving 7 x 5 in image
```

```
ggplot(data = actors %>% filter(company_vs_name != "neither") %>%
         group_by(company_vs_name, updated_days_log) %>%
         summarise(company_vs_name_pct = n()/total_actors, updated_min_max = first(upd
ated_min_max)),
       aes(x=company_vs_name, y=company_vs_name_pct, fill=reorder(updated_min_max, upd
ated_days_log))) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Has Company vs Has name") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete("Last Updated (days)") +
  labs(title="Github Contributors with Profile Company and/or Name")
```




```
ggsave("company_vs_name_last_updated.png")
```

```
## Saving 7 x 5 in image
```

For a sanity check, we look at the company names in the “Company Only” group and find nothing unusual.

```
# company names

actors %>% filter(company_vs_name == "company only") %>% select(company)
```

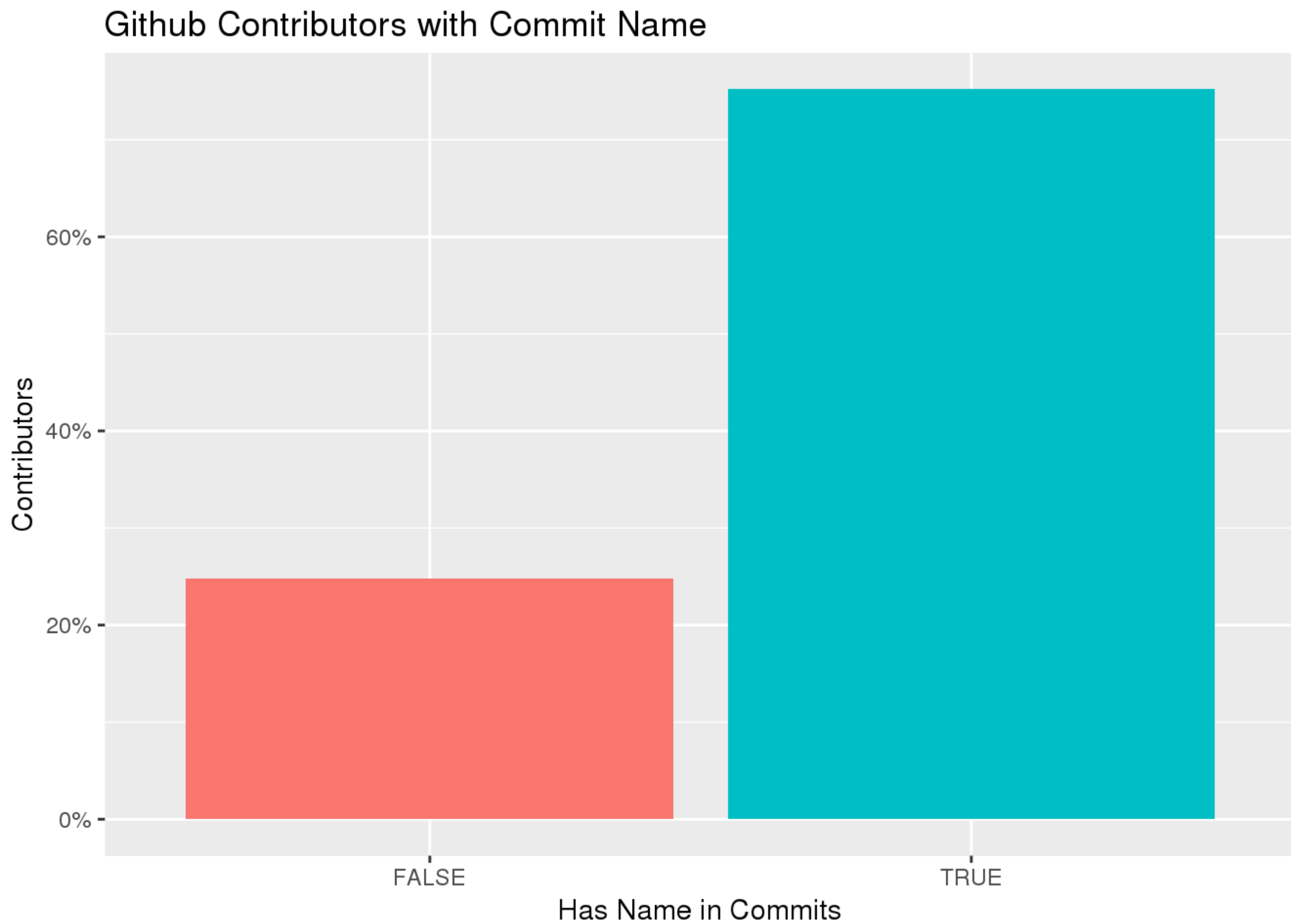
```
##                    company
## 1                     HUST
## 2          Intern @Wingify
## 3                     SJTU
## 4                    opera
## 5                   Amazon
## 6                  Netease
## 7                     SCUT
## 8                  Orbbec
## 9 @BritishGeologicalSurvey
```

Commits Analysis

Names

Almost 70% of contributors had a name in their commit history. We should compare this to the number that had a name in their profile already to see how many of these represent new identifications.

```
# names found in commits
ggplot(data = actors %>% group_by(has_commit_name) %>% summarise(has_commit_name_pct =
 n()/total_actors),
       aes(x=has_commit_name, y=has_commit_name_pct, fill=has_commit_name)) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Has Name in Commits") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide=FALSE) +
  labs(title="Github Contributors with Commit Name")
```

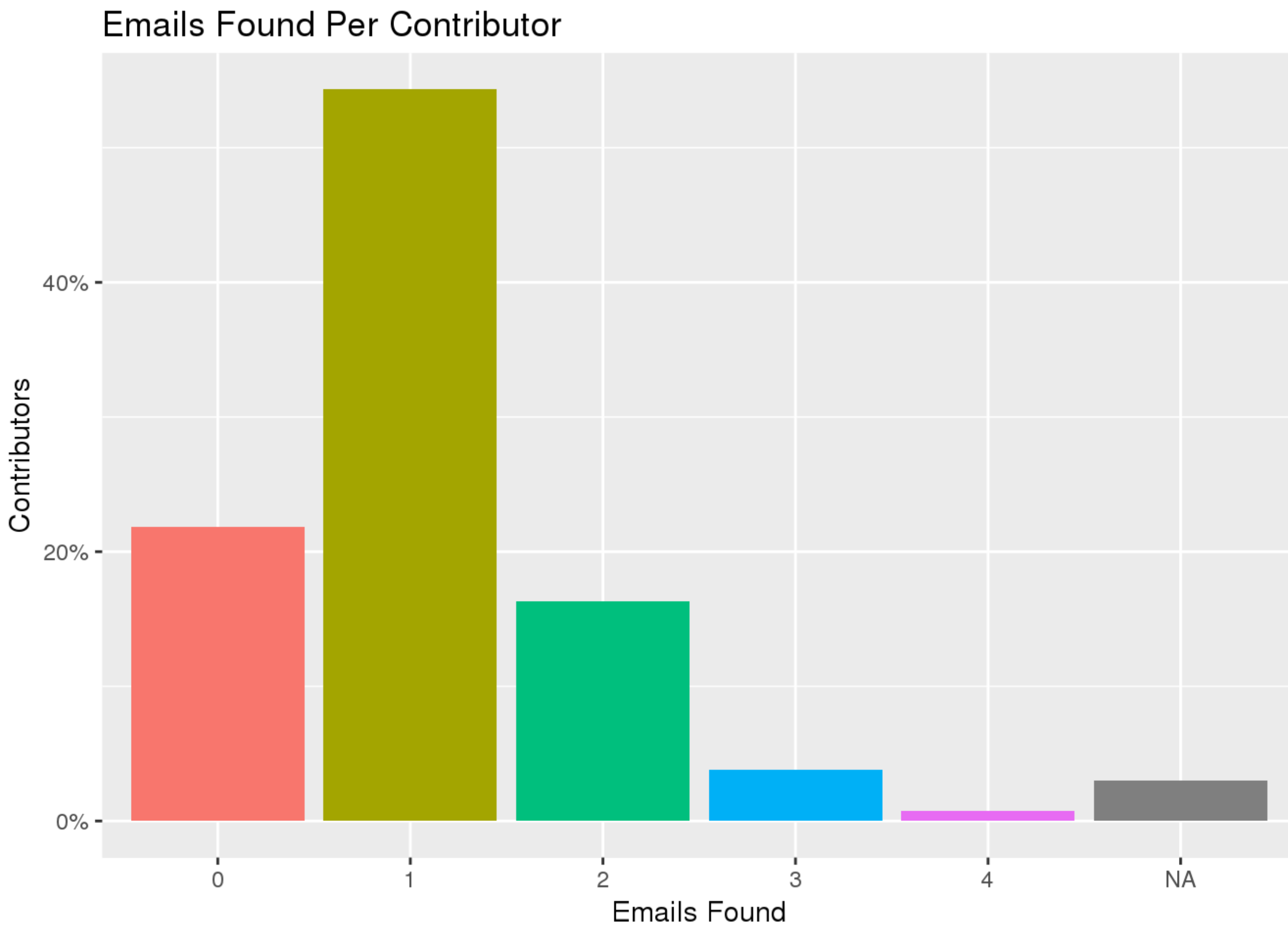



```
ggsave("commit_name.png")
```

```
## Saving 7 x 5 in image
```

Email addresses

```
# email addresses found in commits, overall
ggplot(data = actors %>% group_by(commits_emails_cnt) %>% summarise(commits_emails_pct
 = n()/total_actors),
       aes(x=factor(commits_emails_cnt), y=commits_emails_pct, fill=factor(commits_ema
ils_cnt))) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Emails Found") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide = FALSE) +
  labs(title="Emails Found Per Contributor")
```




```
ggsave("commit_emails.png")
```

```
## Saving 7 x 5 in image
```

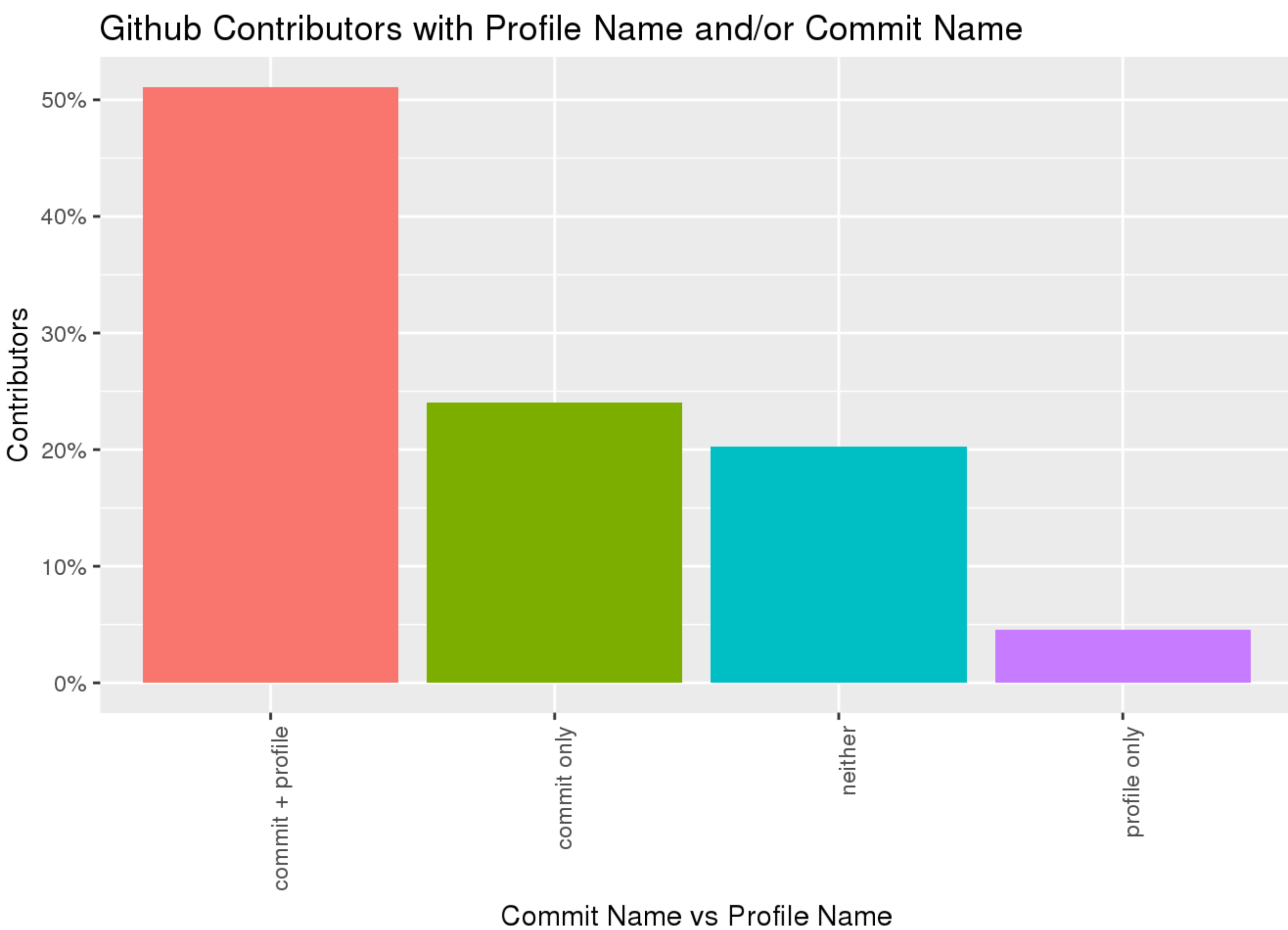
Profiles vs Commits

How successful was the matching? Because we don't get public email addresses from the Github API, we cannot make any conclusions using that field. Names are available in both, however so we'll use that to gauge how successful we were at extracting additional identifying information.

We were able to get names for an additional 25% of contributors that had not provided a name in their profile. By extracting identifying information from commit histories, we are able to potentially identify 80% of the most active mxnet contributors.

The contributors with no information are an interesting case that will be looked at below. Per the following analysis, we could, at most, increase our name extraction by about 4% through improved commit history management.

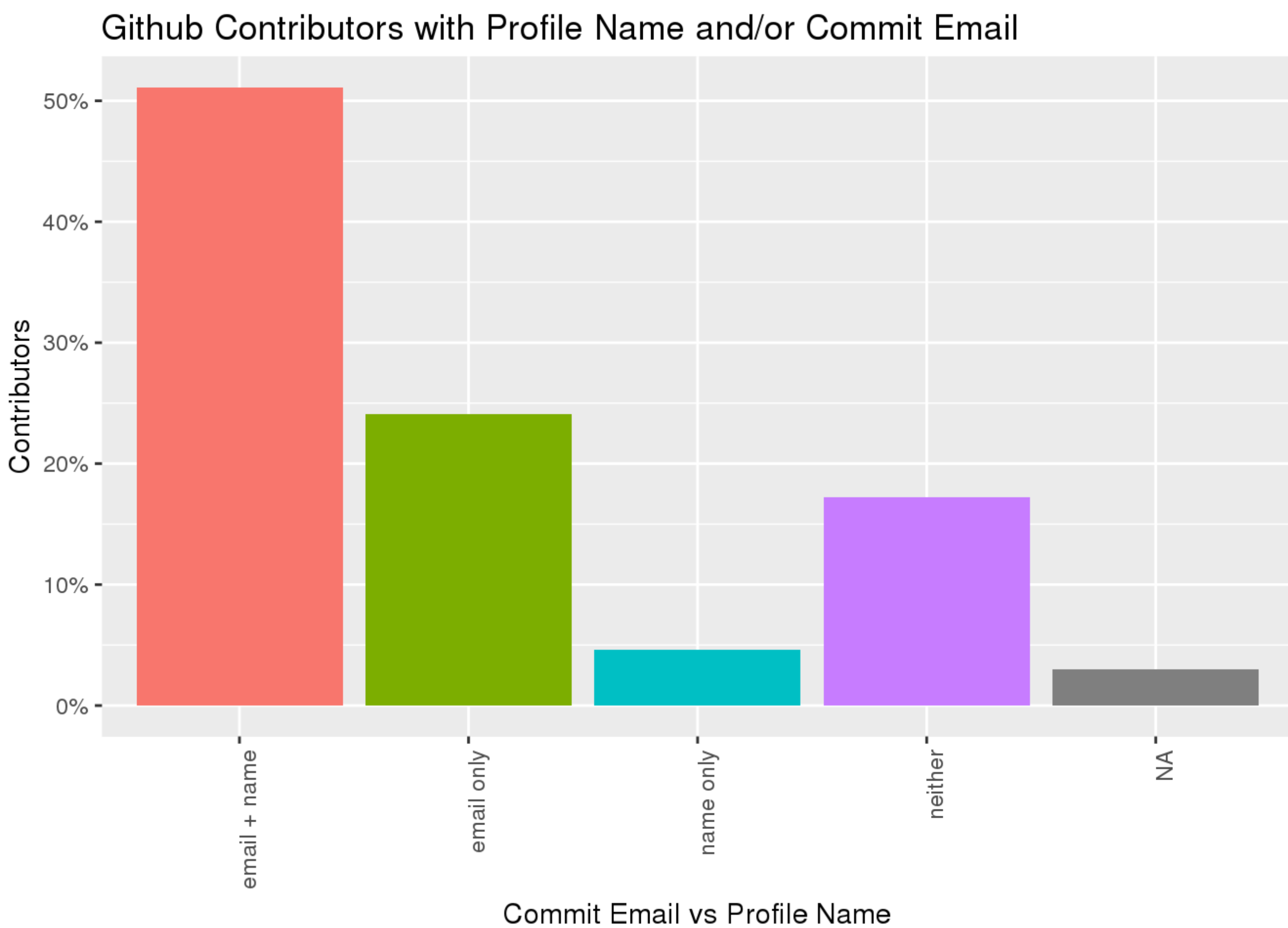
```
# number of actors with name in profile vs number having a name in commit history
actors <- actors %>%
  mutate(name_commit_vs_profile = ifelse(has_commit_name & has_name, paste("commit + p
rofile"),
                                  ifelse(has_commit_name, paste("commit only"),
                                  ifelse(has_name, paste("profile only"),
                                         paste("neither")
                                  )))
  )

ggplot(data = actors %>% group_by(name_commit_vs_profile) %>%
         summarise(name_commit_vs_profile_pct = n()/total_actors),
       aes(x=name_commit_vs_profile, y=name_commit_vs_profile_pct, fill=name_commit_vs
_profile)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Commit Name vs Profile Name") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide=FALSE) +
  labs(title="Github Contributors with Profile Name and/or Commit Name")
```




```
ggsave("name_commit_vs_profile.png")
```

```
## Saving 7 x 5 in image
```

```
actors <- actors %>%
  mutate(email_vs_name = ifelse(has_commit_email & has_name, paste("email + name"),
                                ifelse(has_commit_email, paste("email only"),
                                ifelse(has_name, paste("name only"),
                                       paste("neither")
                                )))
  )

ggplot(data = actors %>% group_by(email_vs_name) %>%
         summarise(email_vs_name_pct = n()/total_actors),
       aes(x=email_vs_name, y=email_vs_name_pct, fill=email_vs_name)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Commit Email vs Profile Name") +
  ylab("Contributors") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide=FALSE) +
  labs(title="Github Contributors with Profile Name and/or Commit Email")
```

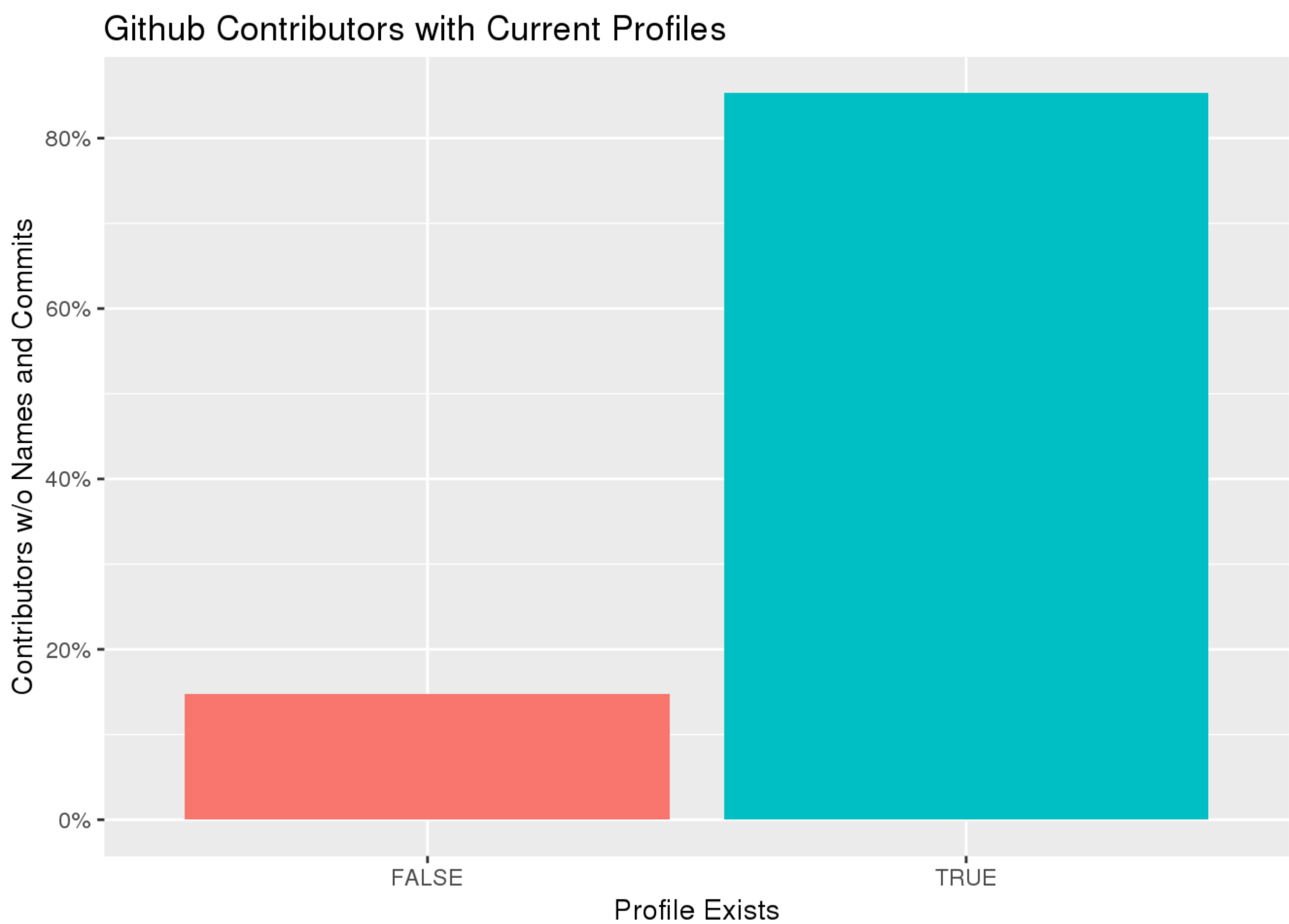


Here we look further at contributors with no names in commits and no name in their profile. Some of the profiles no longer existed when the Github API pull was made. Only a small proportion, ~2.5%, in this group did not have current profiles.

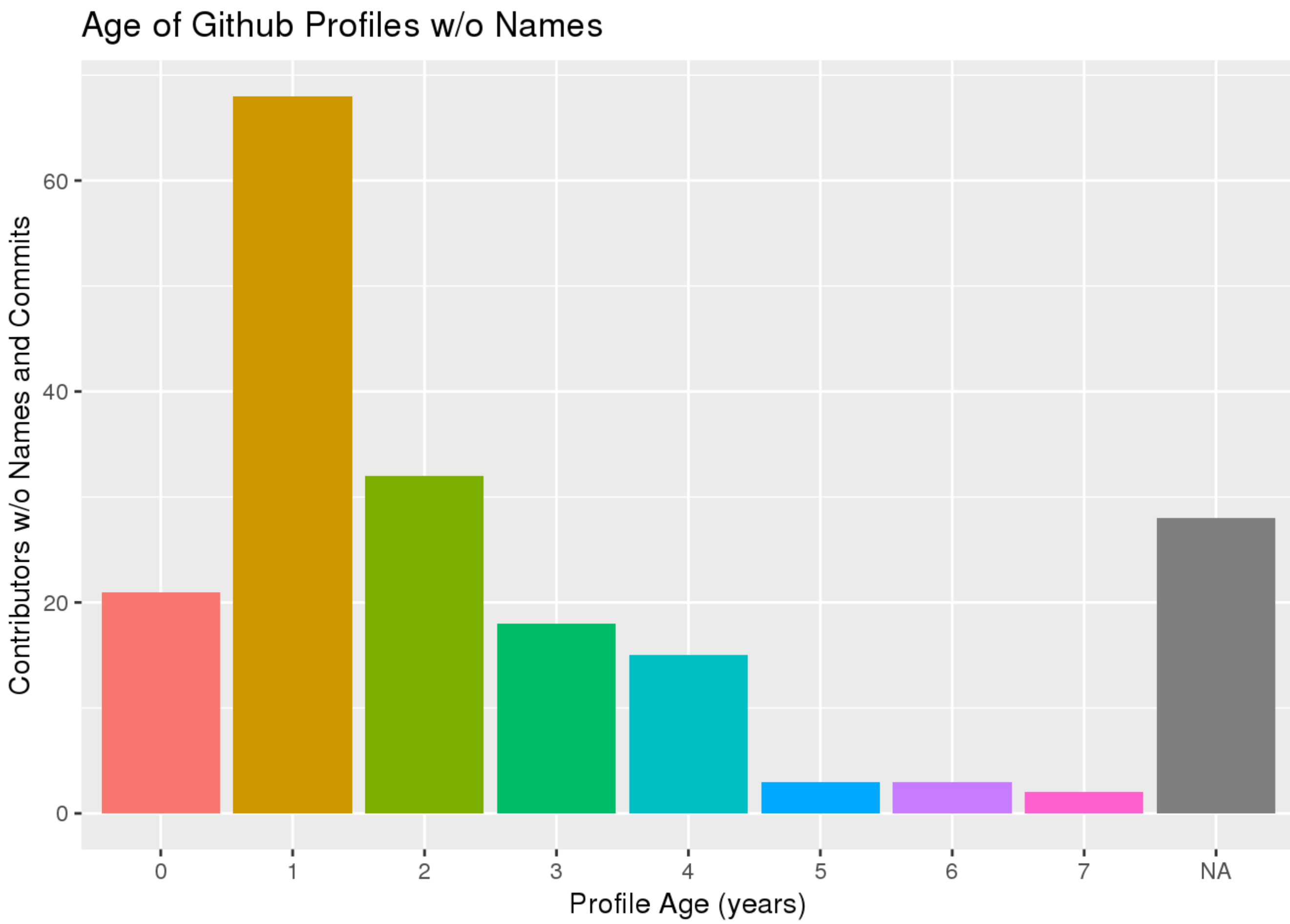
Around 20% of these had a high number of public repos and a manual verification showed that information was available in commit histories depending on how the repositories were sorted in the API request. The script should probably be modified to sort the repos differently. In addition, looking for certain types of events linked to commits in the users' public event stream and extracting the repo name could be another method worth exploring.

The majority of profiles that could not be identified through commit history only zero or just a small number of repositories. Future analysis should consider their event activity in the project that identified them as one of the most active contributors. It's possible refining that metric will reduce these.

```
actors_no_name <- actors %>% filter(name_commit_vs_profile == "neither") %>%
    mutate(current = bio != "404 <Response [404]>" | is.na(bio),
           public_repos_log = round(log(public_repos + 1)))

total_actors_no_name <- nrow(actors_no_name)

ggplot(data = actors_no_name %>% group_by(current) %>% summarise(current_pct = n()/tot
al_actors_no_name),
       aes(x=current, y=current_pct, fill=current)) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Profile Exists") +
  ylab("Contributors w/o Names and Commits") +
  scale_y_continuous(labels = percent) +
  scale_fill_discrete(guide=FALSE) +
  labs(title="Github Contributors with Current Profiles")
```



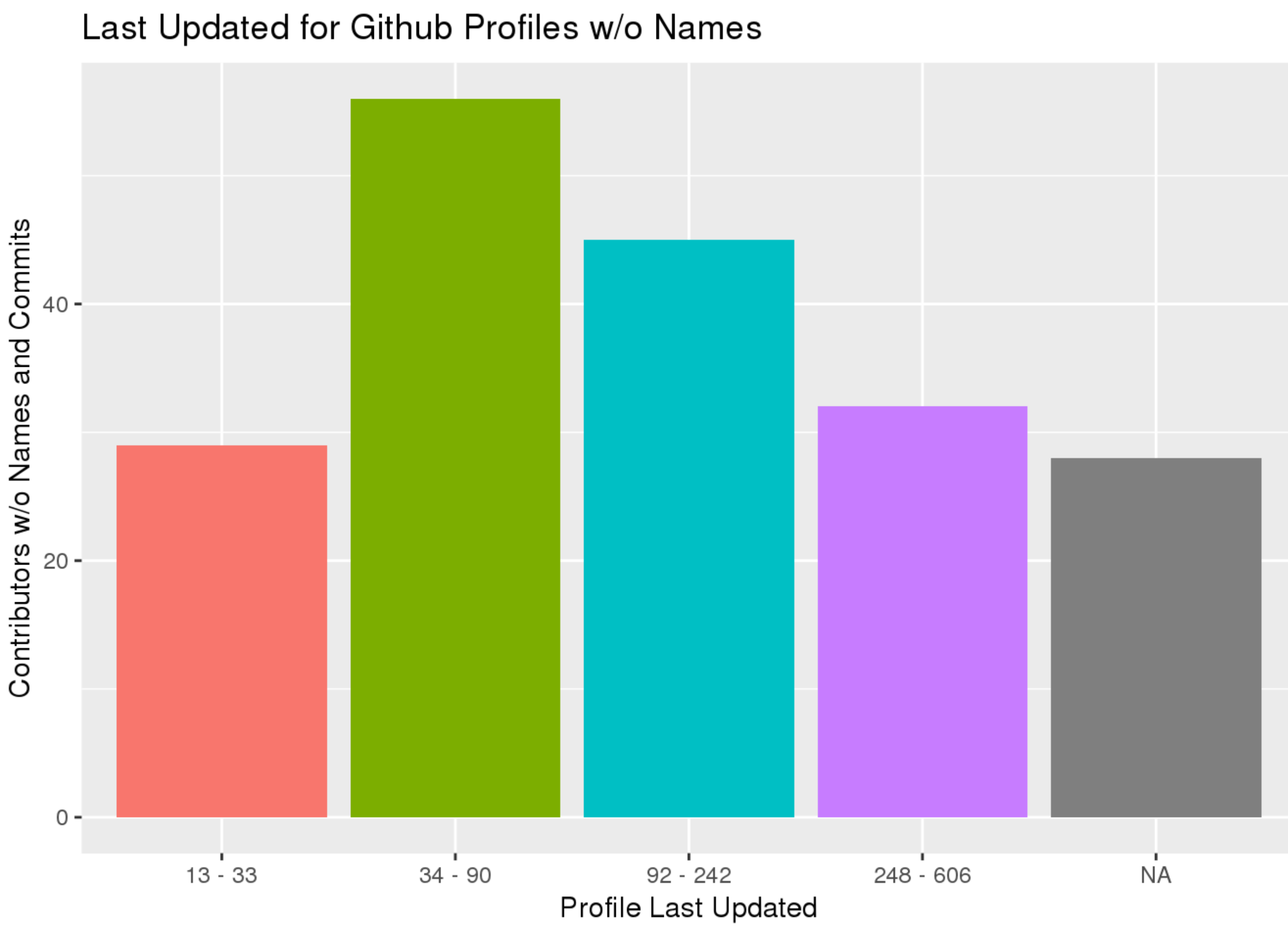
```
# age
ggplot(data = actors_no_name,
       aes(x=factor(age_years), fill=factor(age_years))) +
  geom_bar(position="dodge") +
  scale_fill_discrete(guide = FALSE) +
  xlab("Profile Age (years)") +
  ylab("Contributors w/o Names and Commits") +
  labs(title="Age of Github Profiles w/o Names")
```



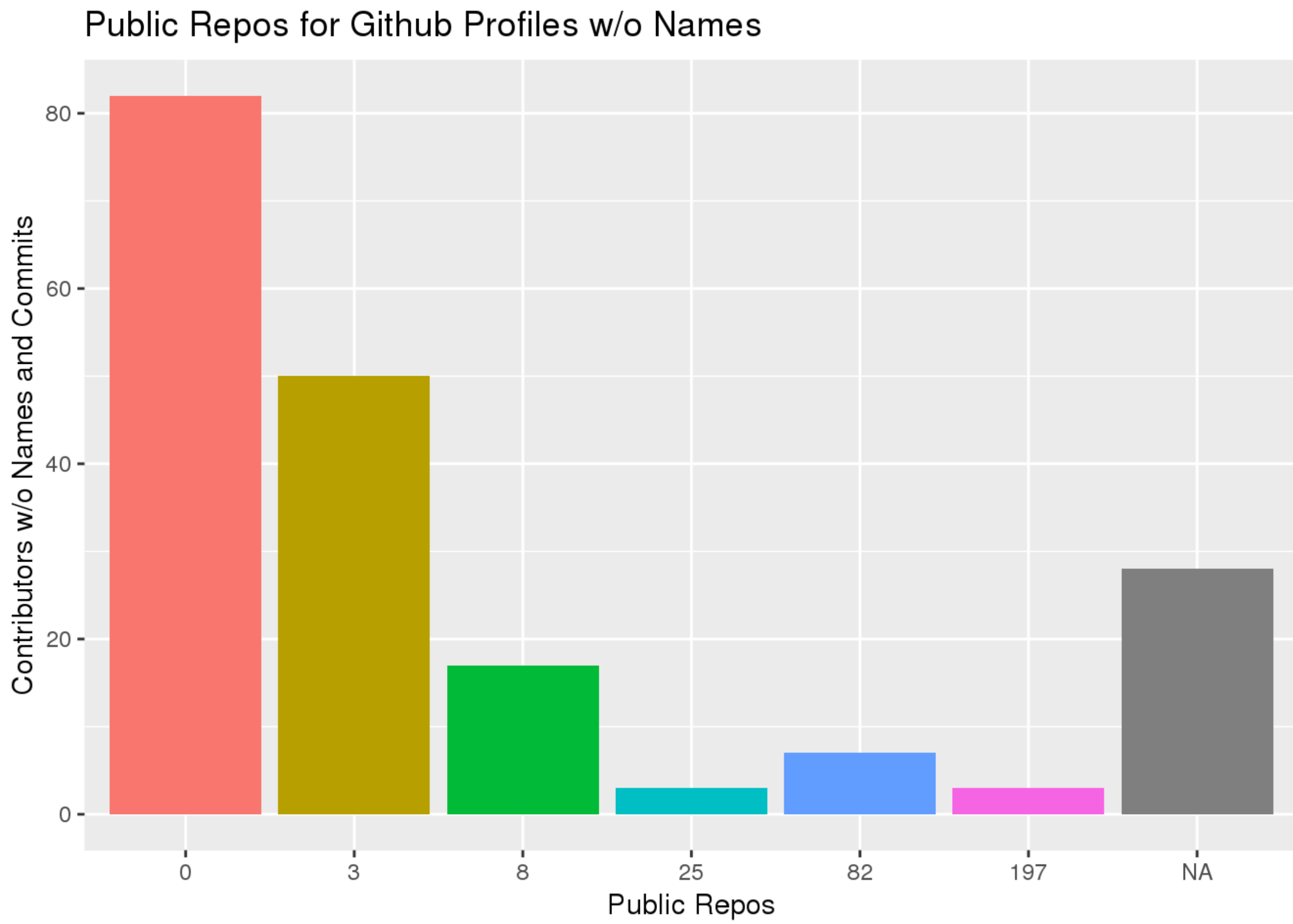

```
ggsave("actors_no_name_commits.png")
```

```
## Saving 7 x 5 in image
```

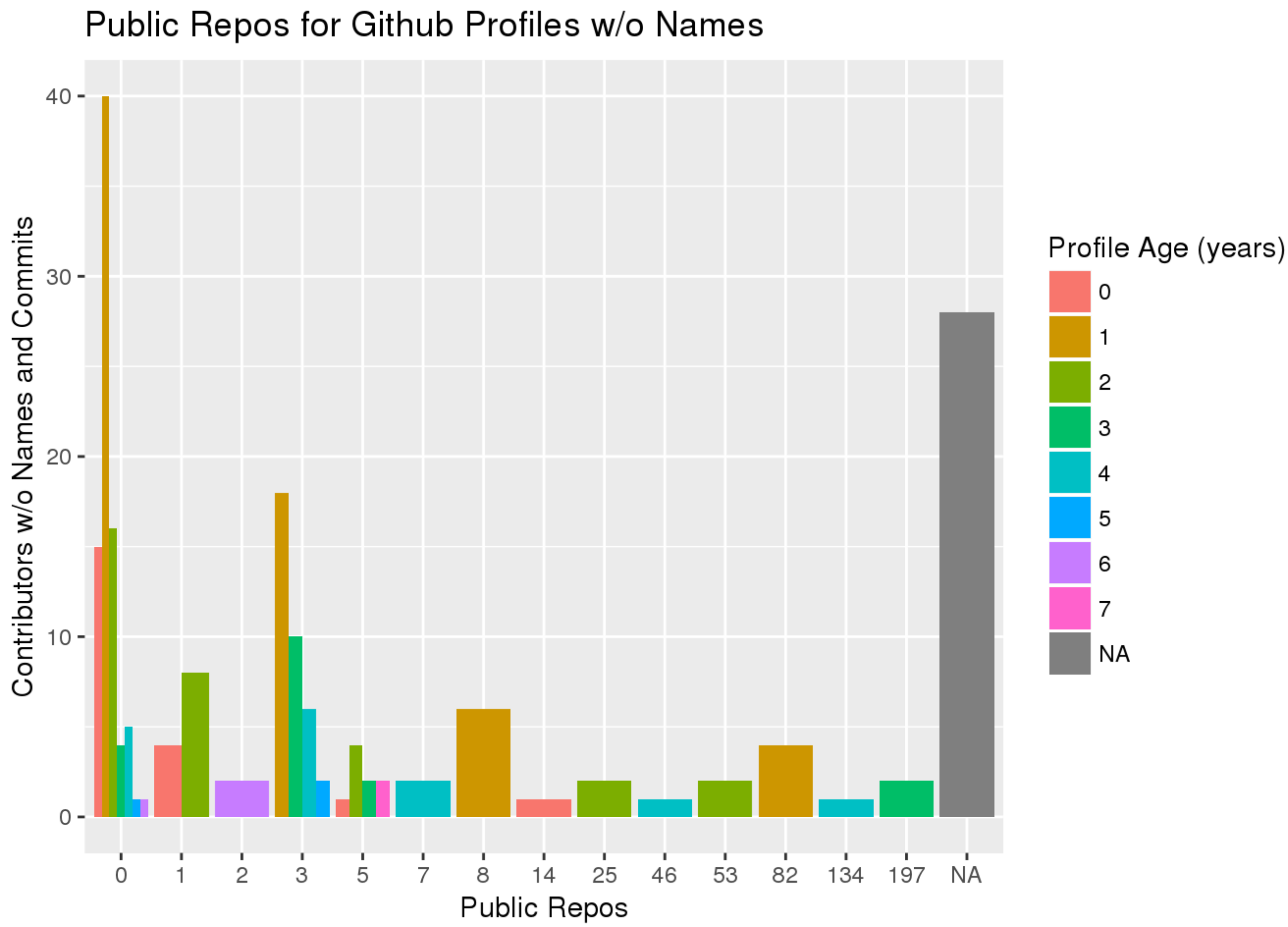
```r
# last updated

ggplot(data = actors_no_name %>% group_by(updated_days_log) %>%
         summarise(updated_days_pct = n(), updated_min_max = first(updated_min_max)),
       aes(x=reorder(updated_min_max, updated_days_log), y=updated_days_pct,
           fill=reorder(updated_min_max, updated_days_log))) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Profile Last Updated") +
  ylab("Contributors w/o Names and Commits") +
  scale_fill_discrete(guide = FALSE) +
  labs(title="Last Updated for Github Profiles w/o Names")
```




```r
ggplot(data = actors_no_name %>% group_by(public_repos_log) %>%
         summarise(public_repos_pct = n(), public_repos=max(public_repos)),
       aes(x=factor(public_repos), y=public_repos_pct, fill=factor(public_repos_log)))
 +
  geom_bar(position="dodge", stat="identity") +
  xlab("Public Repos") +
  ylab("Contributors w/o Names and Commits") +
  scale_fill_discrete(guide = FALSE) +
  labs(title="Public Repos for Github Profiles w/o Names")
```




```
ggplot(data = actors_no_name %>% group_by(public_repos_log, age_years) %>%
         summarise(public_repos_pct = n(),
                   public_repos=max(public_repos)),
       aes(x=factor(public_repos), y=public_repos_pct, fill=factor(age_years))) +
  geom_bar(position="dodge", stat="identity") +
  xlab("Public Repos") +
  ylab("Contributors w/o Names and Commits") +
  scale_fill_discrete("Profile Age (years)") +
  labs(title="Public Repos for Github Profiles w/o Names")
```




```
ggsave("public_repos.png")
```

```
## Saving 7 x 5 in image
```

```
actors_no_name_repos <- actors_no_name %>% filter(public_repos_log > 2)
actors_no_name_repos %>% select(login, company, age_years, public_repos) %>% arrange(d
esc(public_repos))
```

```
##          login company age_years public_repos
## 1    zencoding    <NA>         3          197
## 2      anddelu    <NA>         4          134
## 3    zdltheone    <NA>         3          103
## 4   123chengbo    <NA>         1           82
## 5       Cv9527    <NA>         1           55
## 6        xhniu    <NA>         2           53
## 7       newzhx    <NA>         1           48
## 8    FlyingZXC    <NA>         4           46
## 9  Struggle-YD    <NA>         1           35
## 10    zhyj3038    <NA>         2           33
## 11      morusu    <NA>         2           25
## 12  liuxialong   opera         2           18
## 13  junshipeng    <NA>         0           14
```

Email Address Analysis

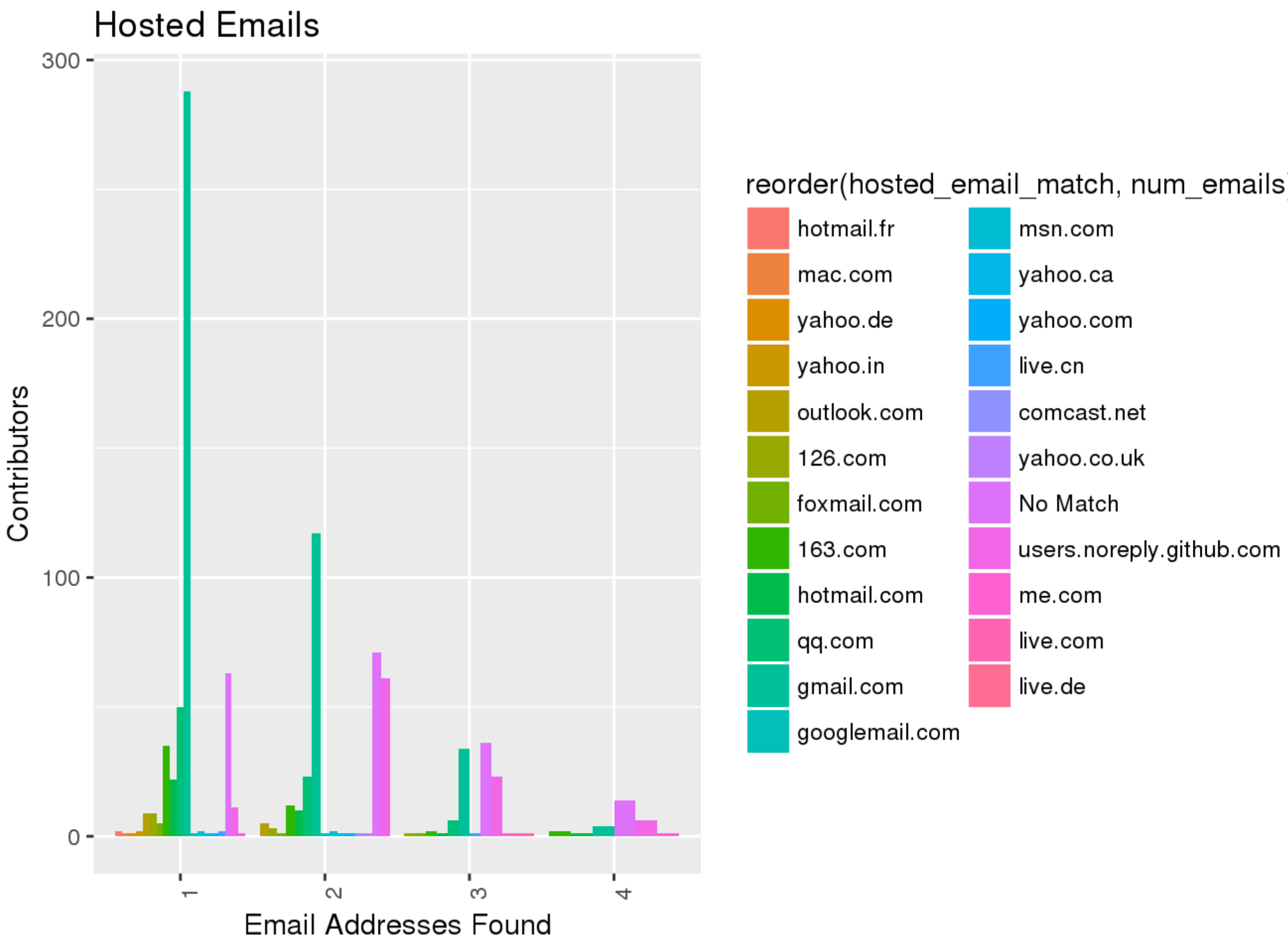
Domains

Domains were extracted from email addresses to compare with company names. To predict the usefulness of these addresses in making company matches, we can identify common domains for hosted email services. We find that the majority of contributors have a hosted email address (eg, gmail.com, hotmail.com) and only one email address. Contributors with more than one email address have a higher chance of having a non-hosted email address. The majority of contributors use gmail.com addresses.

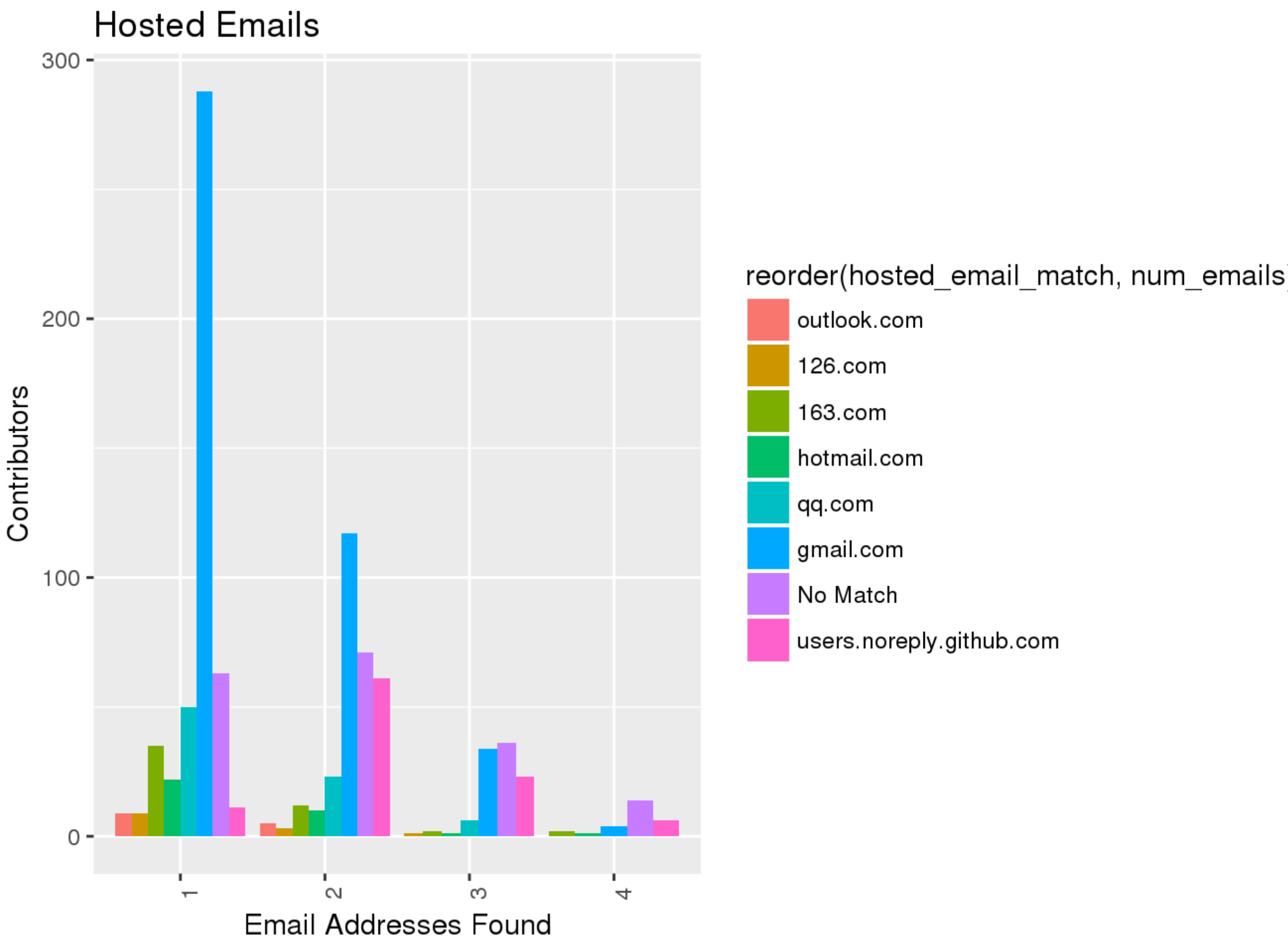
```
# extract domains from email addresses
actors_commit_emails <- actors %>% filter(!is.na(commits_emails)) %>% select(login, co
mmits_emails)

emails <- strsplit(as.character(actors_commit_emails$commits_emails), split = ",")
actors_emails <- data.frame(login = rep(actors_commit_emails$login, sapply(emails, len
gth)),
                            email = unlist(emails))
rm(actors_commit_emails)

actors_emails <- actors_emails %>% mutate(
  host = regmatches(email, regexpr("(?<=@)(.*)", email, perl=TRUE))
)

total_actors_emails <- nrow(actors_emails)

actors_emails_summary <- actors_emails %>%
  group_by(host) %>%
  summarise(emails=n(), emails_pct = round(n()/total_actors_emails, 3)) %>%
  arrange(desc(emails_pct), host)

actors_emails_summary
```

```
## # A tibble: 162 x 3
##                        host emails emails_pct
##                       <chr>  <int>      <dbl>
##  1                gmail.com    443      0.465
##  2 users.noreply.github.com    101      0.106
##  3                   qq.com     78      0.082
##  4                  163.com     51      0.054
##  5              hotmail.com     34      0.036
##  6              outlook.com     14      0.015
##  7                  126.com     13      0.014
##  8               amazon.com      9      0.009
##  9              foxmail.com      7      0.007
## 10                 sina.com      6      0.006
## # ... with 152 more rows
```

```
# what proportion of actors have hosted emails and how many email addresses did they h
ave?

hosted_email = c("gmail.com","users.noreply.github.com","qq.com","163.com", "hotmail.c
om", "outlook.com", "126.com", "foxmail.com", "me.com", "msn.com", "live.cn", "googlem
ail.com", "hotmail.fr", "yahoo.ca", "yahoo.com", "yahoo.in", "comcast.net", "mac.com",
 "live.com", "live.de", "yahoo.de", "yahoo.co.uk")

actors_emails_hosted <- actors_emails %>%
  mutate(hosted_email_match =
           regmatches(host, gregexpr(paste(hosted_email, collapse="|"), host,
                                     perl=TRUE, ignore.case = TRUE))) %>%
  mutate(hosted_email_match =
           ifelse(hosted_email_match == "character(0)", "No Match", paste(hosted_email
_match)),
         hosted_email = hosted_email_match != "No Match")

actors_emails_hosted_summary <- actors_emails_hosted %>%
  group_by(login) %>%
  summarise(
    has_hosted=any(hosted_email),
    num_emails=n()
  )

actors_emails_hosted <- merge(actors_emails_hosted_summary, actors_emails_hosted,
by="login", all=TRUE)

ggplot(data = actors_emails_hosted %>% group_by(login),
       aes(x=factor(num_emails), fill=reorder(hosted_email_match, num_emails))) +
  geom_bar(position="dodge") +
  xlab("Email Addresses Found") +
  ylab("Contributors") +
  labs(title="Hosted Emails") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
actors_emails_summary_top <- actors_emails_summary %>% filter(emails > 9)
ggplot(data = actors_emails_hosted %>%
         filter(hosted_email_match %in% actors_emails_summary_top$host |
                  hosted_email_match == "No Match") %>%
         group_by(login),
       aes(x=factor(num_emails), fill=reorder(hosted_email_match, num_emails))) +
  geom_bar(position="dodge") +
  xlab("Email Addresses Found") +
  ylab("Contributors") +
  labs(title="Hosted Emails") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
ggsave("hosted_emails_top.png")
```

```
## Saving 7 x 5 in image
```

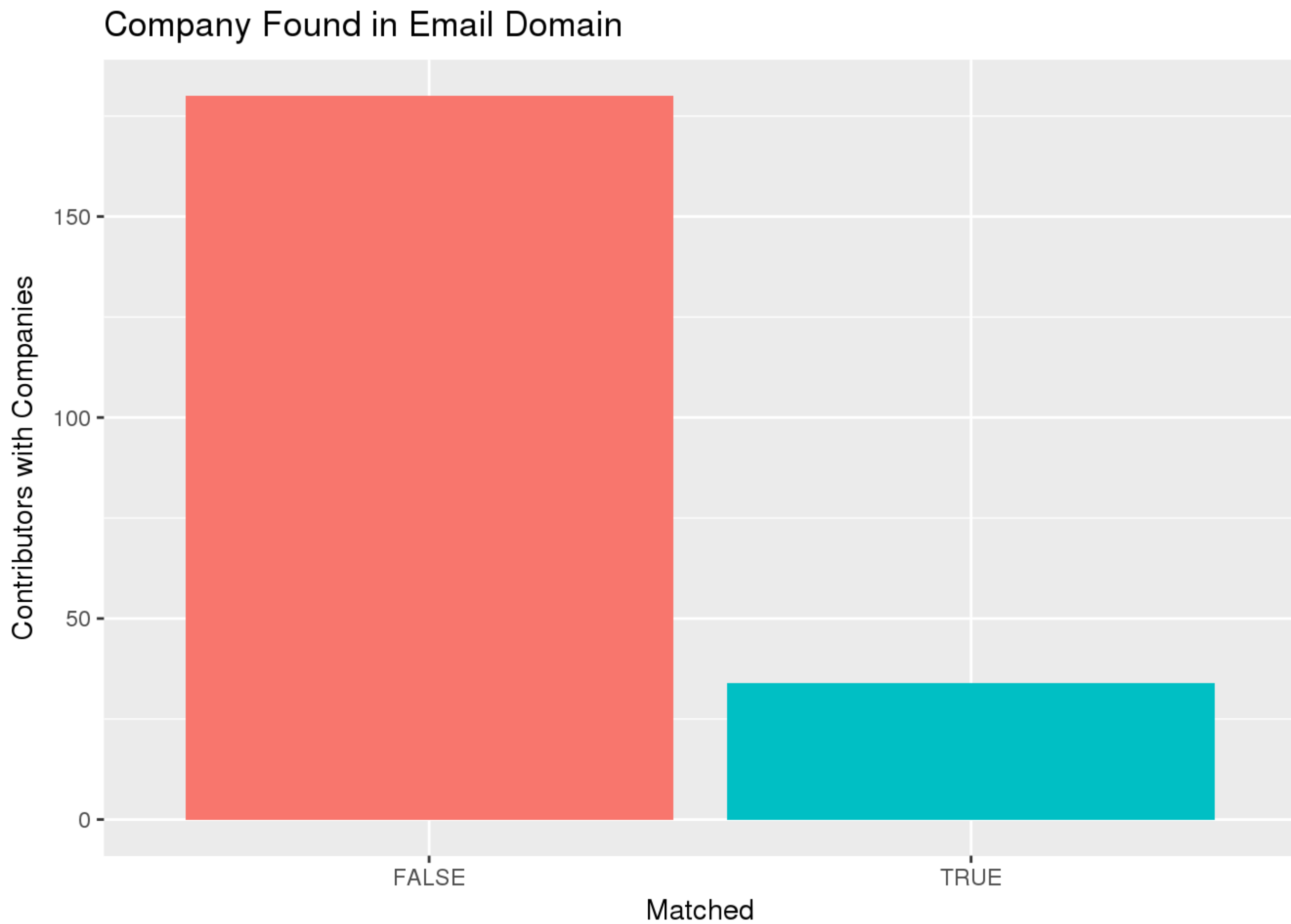
Email Domains vs Company

Can we identify the company from the email domain name? For contributors that have both email addresses in commits and a company name, how well do they match? How does the frequency of company names in profiles compare to the frequency of company-identifiable email addresses?

To make the company_adj field, the company names were manually normalized. This needs to be automated and variations documented, however for now this is sufficient.

Given that most users are using gmail accounts and only have one email address, we should expect this to be pretty low. Less than 20% of normalized company names were found in the email domain. Universities and Corporations showed an equal frequency of domain matches.

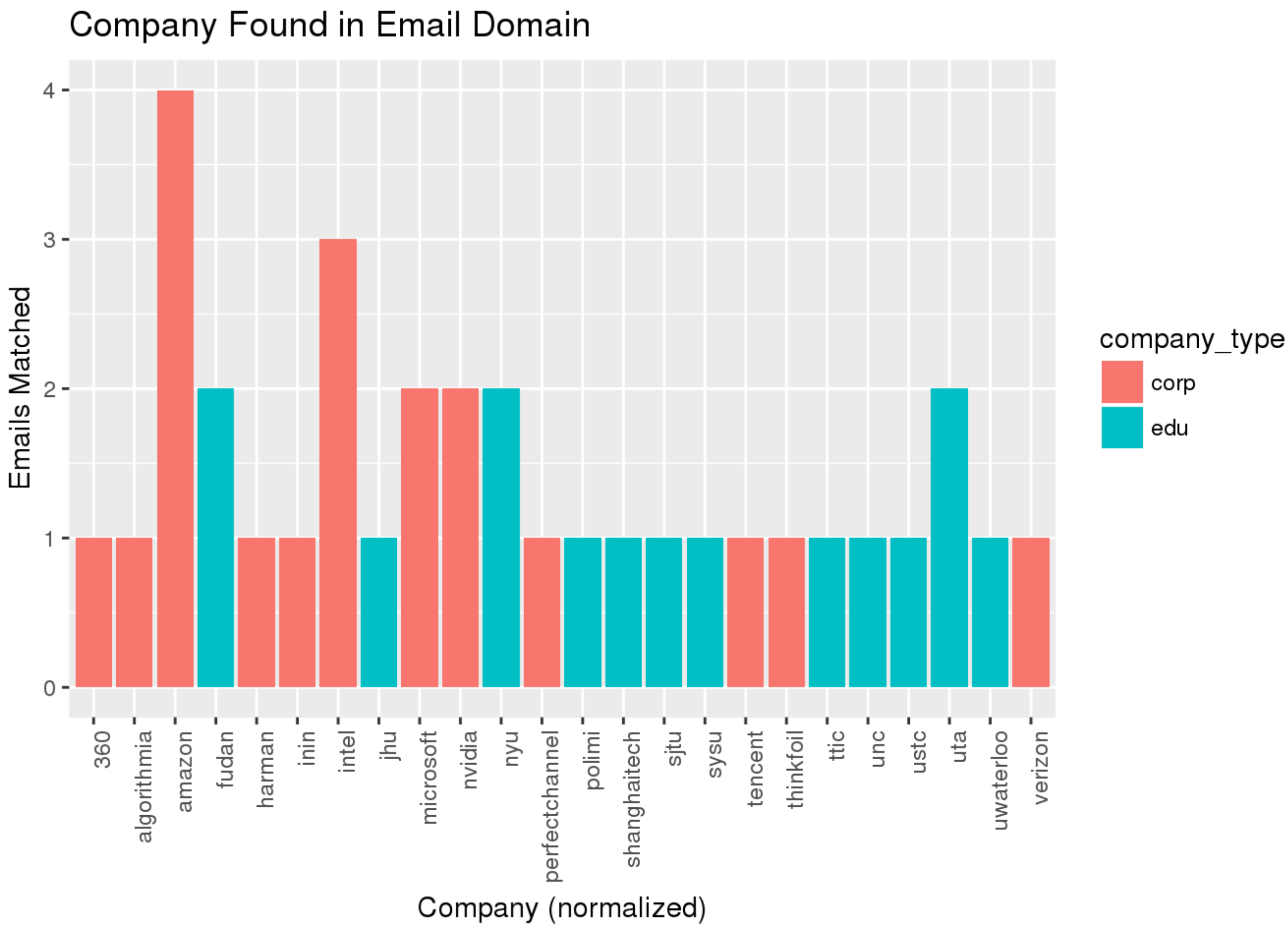
```
actors_companies_adj <- read.csv("mxnet_actors_emails_companies_adj.csv",
na.strings="")

actors_companies_adj$company_in_email <- mapply(grepl,
                                                pattern=actors_companies_adj$company_a
dj,
                                                x=actors_companies_adj$host,
                                                fixed=TRUE)

actors_companies_summary <- actors_companies_adj %>%
  filter(!is.na(company_in_email)) %>%
  group_by(login) %>%
  summarise(company_in_email=any(company_in_email),
            company_adj=first(company_adj),
            company_type=first(company_type))

total_actors_companies <- nrow(actors_companies_summary)

ggplot(data = actors_companies_summary %>% group_by(company_in_email) %>%
         summarise(company_pct = n()),
       aes(x=company_in_email, y=company_pct, fill=company_in_email)) +
  geom_bar(position="dodge", stat="identity") +
  scale_fill_discrete(guide = FALSE) +
  xlab("Matched") +
  ylab("Contributors with Companies") +
  labs(title="Company Found in Email Domain")
```




```
ggsave("company_match_with_company.png")
```

```
## Saving 7 x 5 in image
```

```
ggplot(data = actors_companies_summary %>% group_by(company_adj) %>%
         summarise(num_actors=n(),
                   matched=sum(company_in_email),
                   matched_pct=matched/num_actors,
                   company_type=first(company_type))
       %>% filter(matched > 0),
       aes(x=company_adj, y=matched, fill=company_type)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Company (normalized)") +
  ylab("Emails Matched") +
  labs(title="Company Found in Email Domain")
```




```
ggsave("company_match_with_company_hosts.png")
```

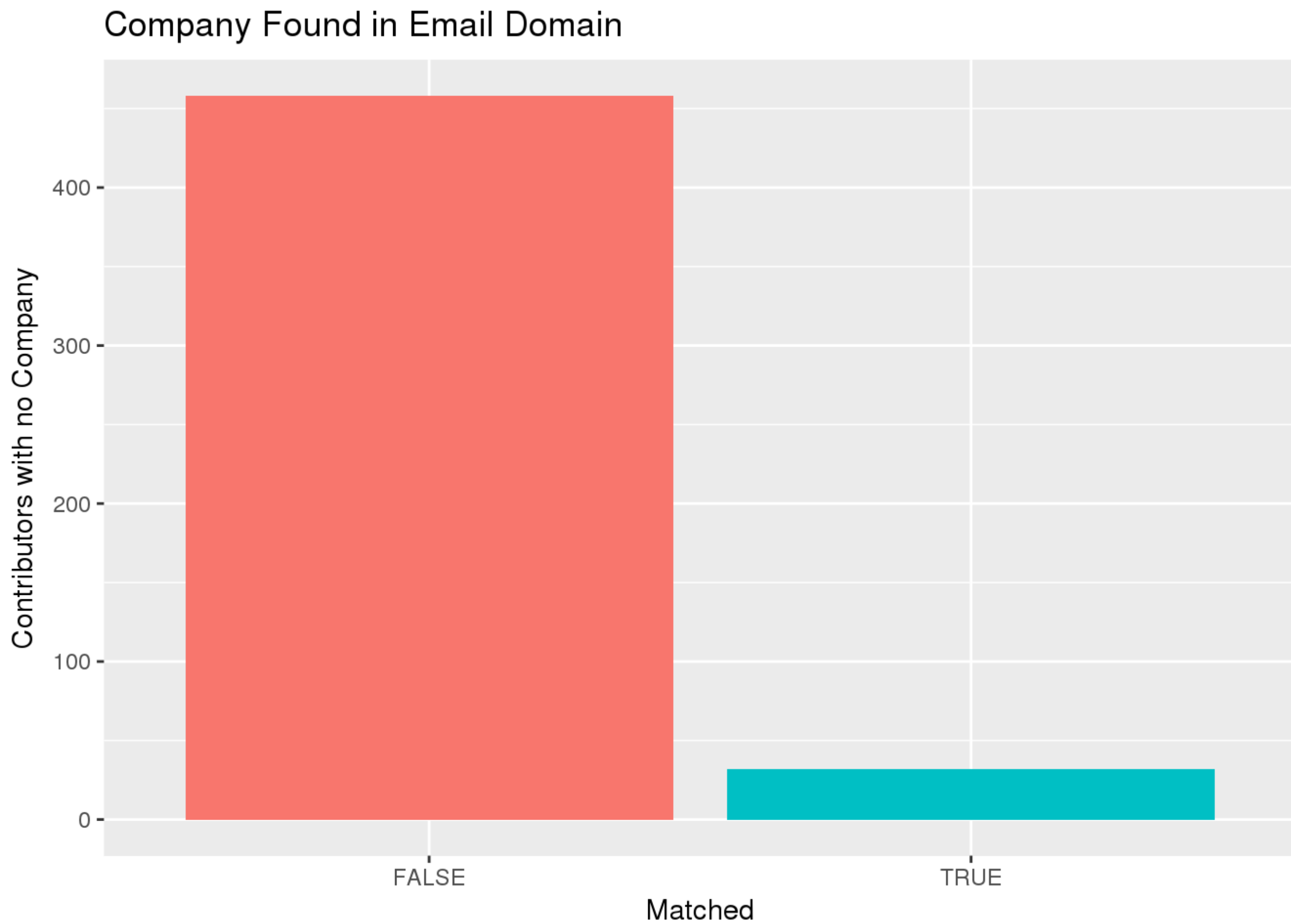
```
## Saving 7 x 5 in image
```

The above analysis suggests we may be able to find additional company matches by checking email domains against a list of normalized company names. Less than 10% of the Github profiles without companies were matched. The method used for this was very simple, the normalized company names found above were checked against the email domains. This depends on the Company name normalization matching their email domain name, and the Github user having committed under their work email address. This could pick up old employers or there could be a false match if the company name is an acronym or very short.

The highest single number of matches came from Amazon, but Amazon is the most represented company in this project, so that may be reflecting that skew. Further analysis is needed on other projects to see if something similar is reflected.

In this sample most of the matches actually came from universities so it could be a reasonable way of identifying contributors affiliated with education (either past or present).

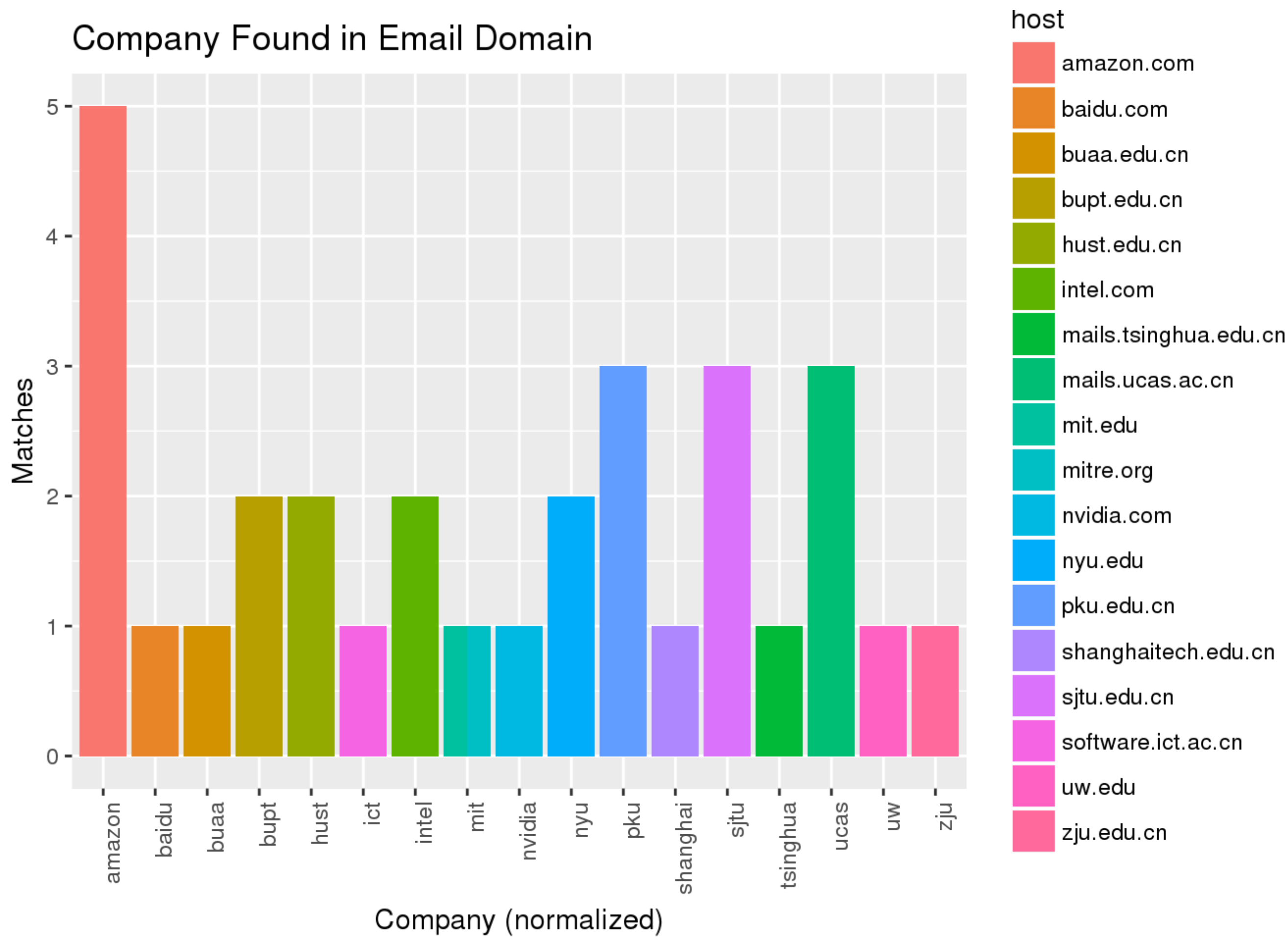
```
# list of normalized company names
companies <- actors_companies_adj %>% filter(!is.na(company_adj)) %>%
  group_by(company_adj) %>%
  summarise(matched=sum(company_in_email), num_actors=n(), matched_pct=round(matched/n
um_actors,2))
companies_str <- paste(companies$company_adj, collapse="|")

# list of contributors with emails that matched
no_company <- actors_companies_adj %>% filter(is.na(company_adj)) %>%
  mutate(company_match = regmatches(host, gregexpr(companies_str, host, perl=TRUE, ign
ore.case = TRUE))) %>%
  mutate(company_match = ifelse(company_match == "character(0)", "No Match", paste(com
pany_match)),
         company_in_email = company_match != "No Match")

logins_matched <- no_company %>%
  filter(company_in_email) %>%
  mutate(has_match=TRUE, email_matched=email, host_matched=host,
         company_matched=company_match, company_matched_type=company_type) %>%
  select(login, has_match, email_matched, host_matched, company_matched, company_match
ed_type)

emails_matched <- merge(logins_matched, no_company, by="login", all=TRUE, incomparable
s=NA)

emails_matched_summary <- emails_matched %>%
  group_by(login) %>%
  summarise(
    num_emails = n(),
    email=first(email_matched),
    host=first(host_matched),
    company_match = first(company_matched),
    company_in_email=any(company_in_email),
    company_type = first(company_matched_type))

actors_no_company <- no_company %>% group_by(login) %>% summarise()
total_actors_no_company <- nrow(actors_no_company)

ggplot(data = emails_matched_summary %>% group_by(company_in_email) %>%
         summarise(company_pct = n()),
       aes(x=company_in_email, y=company_pct, fill=company_in_email)) +
  geom_bar(position="dodge", stat="identity") +
  scale_fill_discrete(guide = FALSE) +
  xlab("Matched") +
  ylab("Contributors with no Company") +
  labs(title="Company Found in Email Domain")
```




```
ggsave("company_match_no_company.png")
```

```
## Saving 7 x 5 in image
```

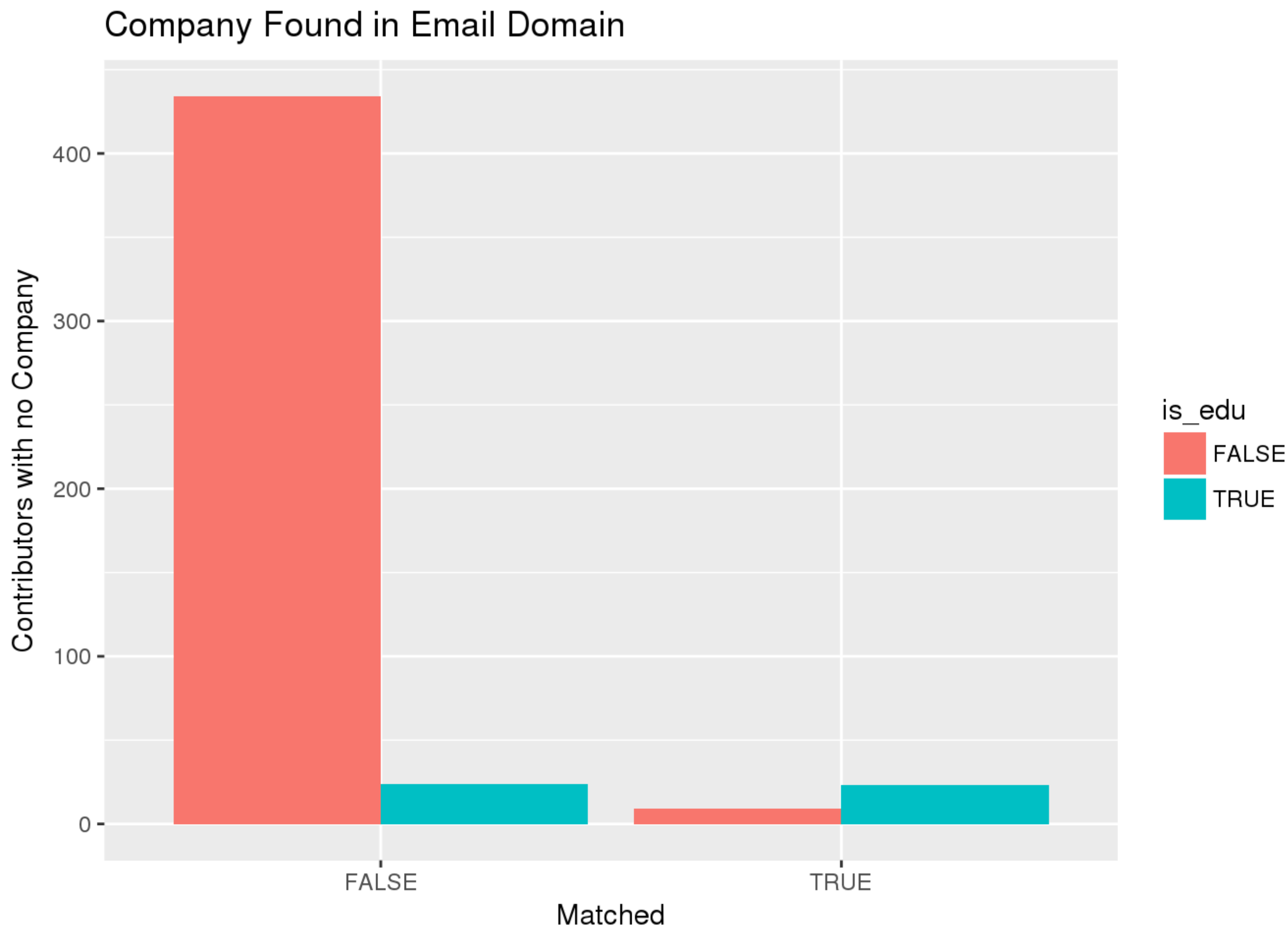
```
ggplot(data = emails_matched_summary %>% filter(!is.na(company_match)),
       aes(x=company_match, fill=host)) +
  geom_bar(position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("Company (normalized)") +
  ylab("Matches") +
  labs(title="Company Found in Email Domain")
```




```
ggsave("company_match_no_company_hosts.png")
```

```
## Saving 7 x 5 in image
```

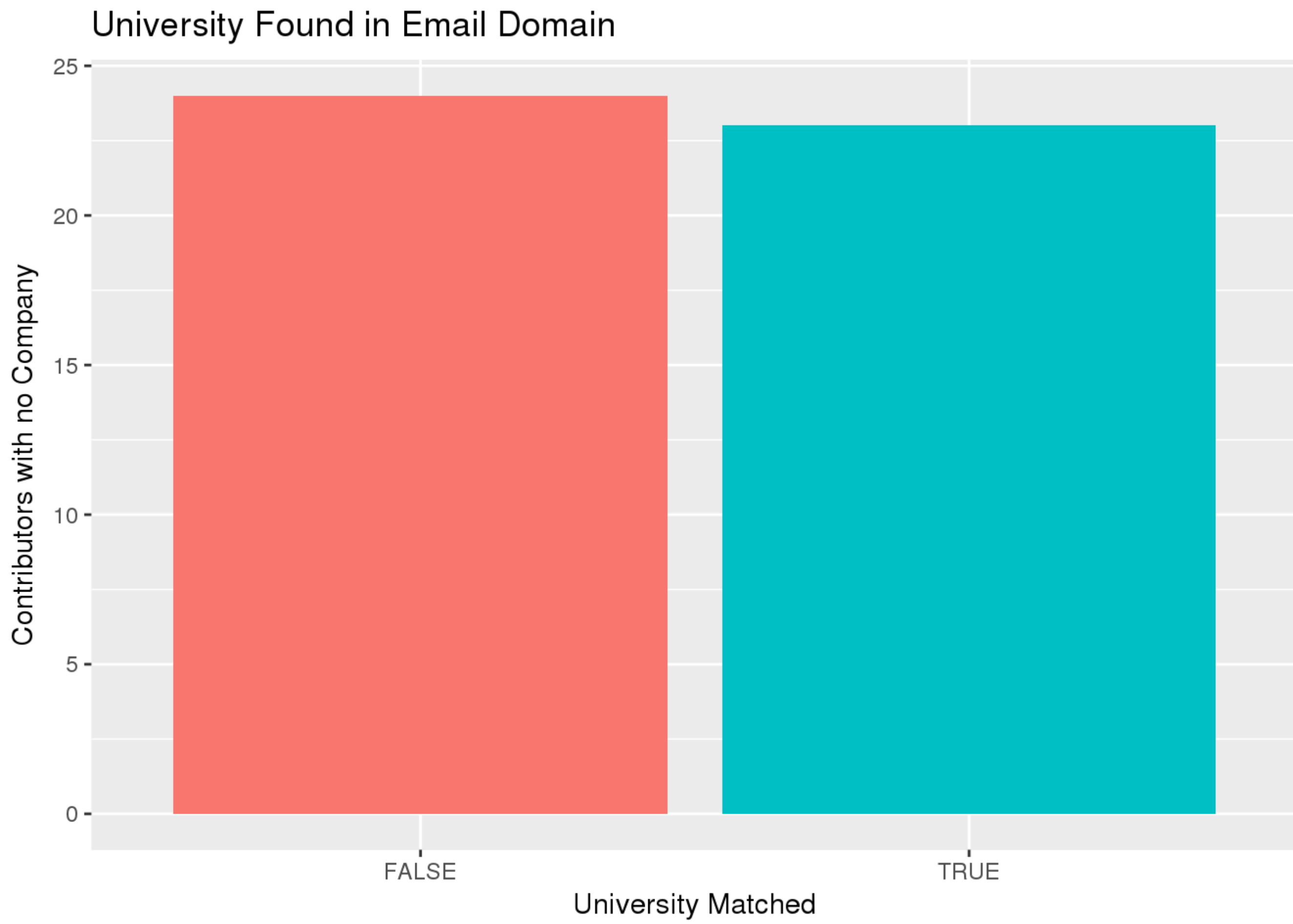
```
# % of edu that matched (not perfect, eu is harder to tell from just domain)

no_company_edu <- no_company %>% filter(is.na(company_adj)) %>%
  mutate(edu_match = regmatches(host, gregexpr('\\.edu|\\.ac\\.', host, perl=TRUE, ign
ore.case = TRUE))) %>%
  mutate(edu_match = ifelse(edu_match == "character(0)", "No Match",
paste(edu_match)),
         is_edu = edu_match != "No Match")

logins_matched_edu <- no_company_edu %>%
  filter(is_edu) %>%
  mutate(has_match=TRUE, email_matched=email, host_matched=host, company_matched=compa
ny_match,
         edu_matched=edu_match, has_edu=TRUE) %>%
  select(login, has_match, email_matched, host_matched, company_matched, edu_matched,
has_edu)

emails_matched_edu <- merge(logins_matched_edu, no_company_edu, by="login", all=TRUE,
incomparables=NA)

emails_matched_edu_summary <- emails_matched_edu %>%
  group_by(login) %>%
  summarise(
    num_emails = n(),
    email=first(email_matched),
    host=first(host_matched),
    company_match = first(company_matched),
    company_in_email=any(company_in_email),
    is_edu = any(has_edu),
    edu_match=first(edu_matched)
) %>%
  mutate(
    is_edu = ifelse(is.na(is_edu), FALSE, is_edu)
  )

ggplot(data = emails_matched_edu_summary,
       aes(x=company_in_email, fill=is_edu)) +
  geom_bar(position="dodge") +
  #scale_fill_discrete(guide = FALSE) +
  xlab("Matched") +
  ylab("Contributors with no Company") +
  labs(title="Company Found in Email Domain")
```



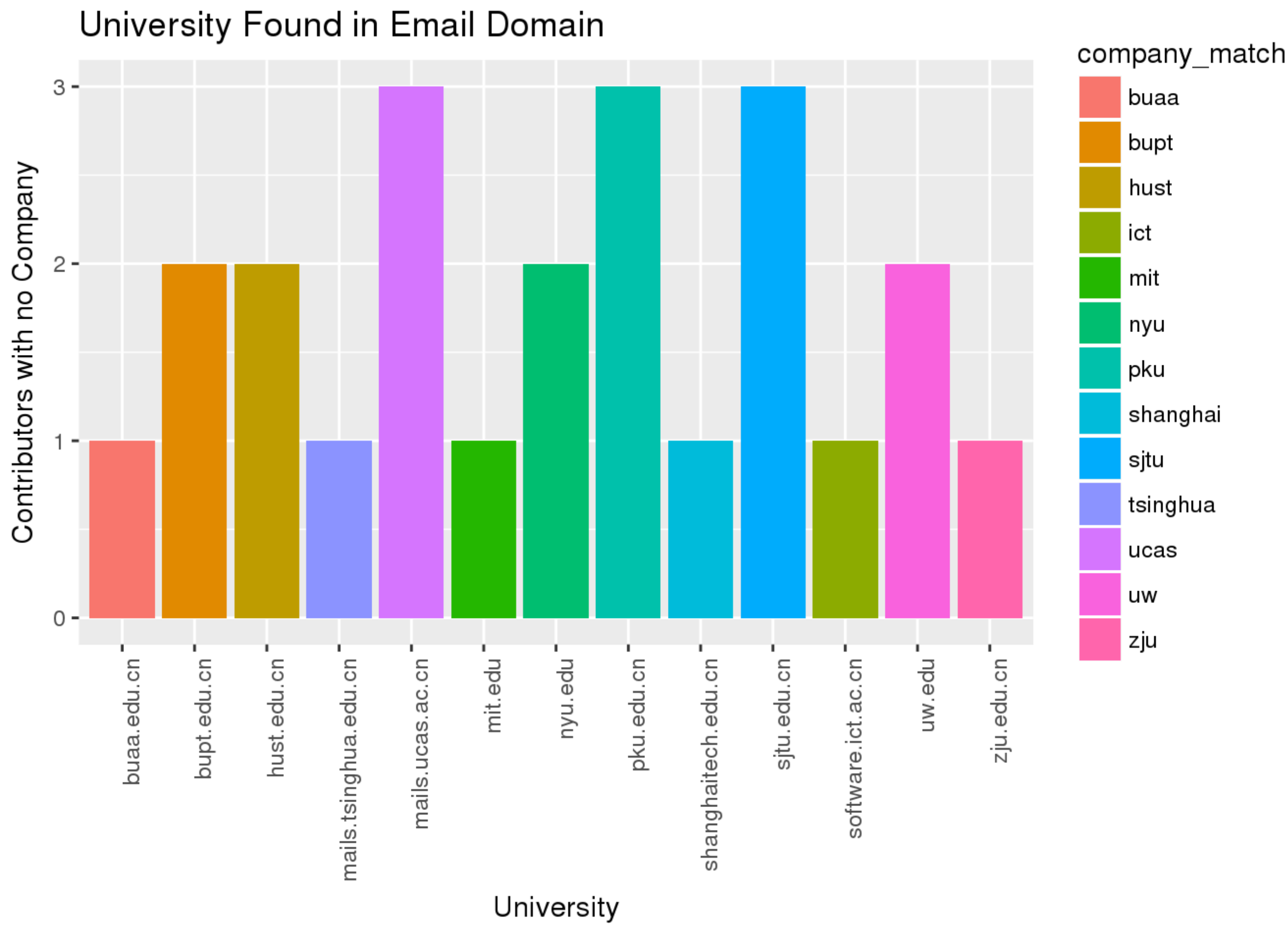

```
ggsave("edu_matched.png")
```

```
## Saving 7 x 5 in image
```

```
ggplot(data = emails_matched_edu_summary %>% filter(is_edu),
       aes(x=company_in_email, fill=company_in_email)) +
  geom_bar(position="dodge") +
  scale_fill_discrete(guide = FALSE) +
  xlab("University Matched") +
  ylab("Contributors with no Company") +
  labs(title="University Found in Email Domain")
```




```
ggplot(data = emails_matched_edu_summary %>% filter(is_edu & company_in_email),
       aes(x=host, fill=company_match)) +
  geom_bar(position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  xlab("University") +
  ylab("Contributors with no Company") +
  labs(title="University Found in Email Domain")
```




```
ggsave("edu_domains_found.png")
```

```
## Saving 7 x 5 in image
```

```
emails_matched_edu_summary %>% filter(is_edu & !company_in_email) %>% select(host)
```

```
## # A tibble: 24 x 1
##                    host
##                  <fctr>
##  1 college.harvard.edu
##  2           cqu.edu.cn
##  3              uci.edu
##  4    life.hkbu.edu.hk
##  5       postech.ac.kr
##  6              aus.edu
##  7   cs.washington.edu
##  8        eng.ucsd.edu
##  9          ntu.edu.tw
## 10            umbc.edu
## # ... with 14 more rows
```

```
no_match_edu <- emails_matched_edu_summary %>% filter(is_edu & !company_in_email)
paste(no_match_edu$host)
```

```
##  [1] "college.harvard.edu" "cqu.edu.cn"          "uci.edu"
##  [4] "life.hkbu.edu.hk"    "postech.ac.kr"       "aus.edu"
##  [7] "cs.washington.edu"   "eng.ucsd.edu"        "ntu.edu.tw"
## [10] "umbc.edu"            "shu.edu.cn"          "mails.jlu.edu.cn"
## [13] "i2r.a-star.edu.sg"   "mail.wbs.ac.uk"      "whu.edu.cn"
## [16] "usc.edu"             "duke.edu"            "buffalo.edu"
## [19] "unist.ac.kr"         "umich.edu"           "ucdavis.edu"
## [22] "stu.xmu.edu.cn"      "mail.bnu.edu.cn"     "psu.edu"
```

Conclusions

Names and Email addresses can be extracted from a user's commit history

The proposed hypothesis that emails could be extracted from commit history is true. Emails and names were extracted for the majority of top contributors in the project.

Company can sometimes be identified from the email domain name

Overall the rate of identification for company from domain was very low. The majority of emails extracted were from hosted email providers, not employers. Further steps need to be taken with this email in order to attempt a company identification.

The majority of successful identifications were university domains.

Normalizing company names based on their web domains and comparing against email addresses would improve domain matches but not significantly. Universities in particular would show an increase in match frequency using this method.

Next Steps

Because the majority of email addresses came from hosted email accounts, we should explore methods of entity resolution to identify company affiliation.