# Participation Rate based on Github Repository Public Events

## Overview

The purpose of this study is to determine if we can predict the overall participation level for a repository based on public Github event activity. The ultimate goal is to develop a sampling methodology for Github repositories based on this event activity.

This study is part of a larger research project to answer the following questions:

- What are the measurable impacts of Continuous Integration?
- What software projects would most likely benefit from Continuous Integration?

## Hypotheses

Participation rate (the proportion of events per actor based on events data) provides an accurate metric for categorizing GitHub repositories.

Stratifying GitHub repositories by participation rate improves analysis results by reducing variability.

Some event types have a higher frequency of certain participation rates.

## Null Hypothesis

Calculating participation rate from events data does not accurately reflect the actual participation rate of the repository.

Grouping repositories by participation does not significantly reduce variability.

Event types do not favor any participation rate.

## Methodology

This research analyzed 6 months of public GitHub activity published in the GitHub Archive, from June 2016 to December 2016. For data set creation, see the accompanying file, “events_analysis_data.Rmd”.

```
library(dplyr)
```

```
##
## 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
```

```
library(ggplot2)
library(reshape2)
```

# Measuring Participation

In the previous section, I identified types of events that occurred more frequently than others. However the total number of actors as well as the actors to repository ratio suggest these events may be associated with repositories that have a small number of contributors. In this section, I'll explore measuring repository participation and grouping repositories by this participation rating.

The repositories represented by the GitHub event data vary significantly. Some have only 1 or 2 contributors while others have several hundred. A normalized metric is required to effectively compare these repositories. The average percent of events per actor can provide a way to group repositories by a comparable participation level.

percent of events per actor = events per actor / total number of events where events per actor = total number of events/total number of unique actors

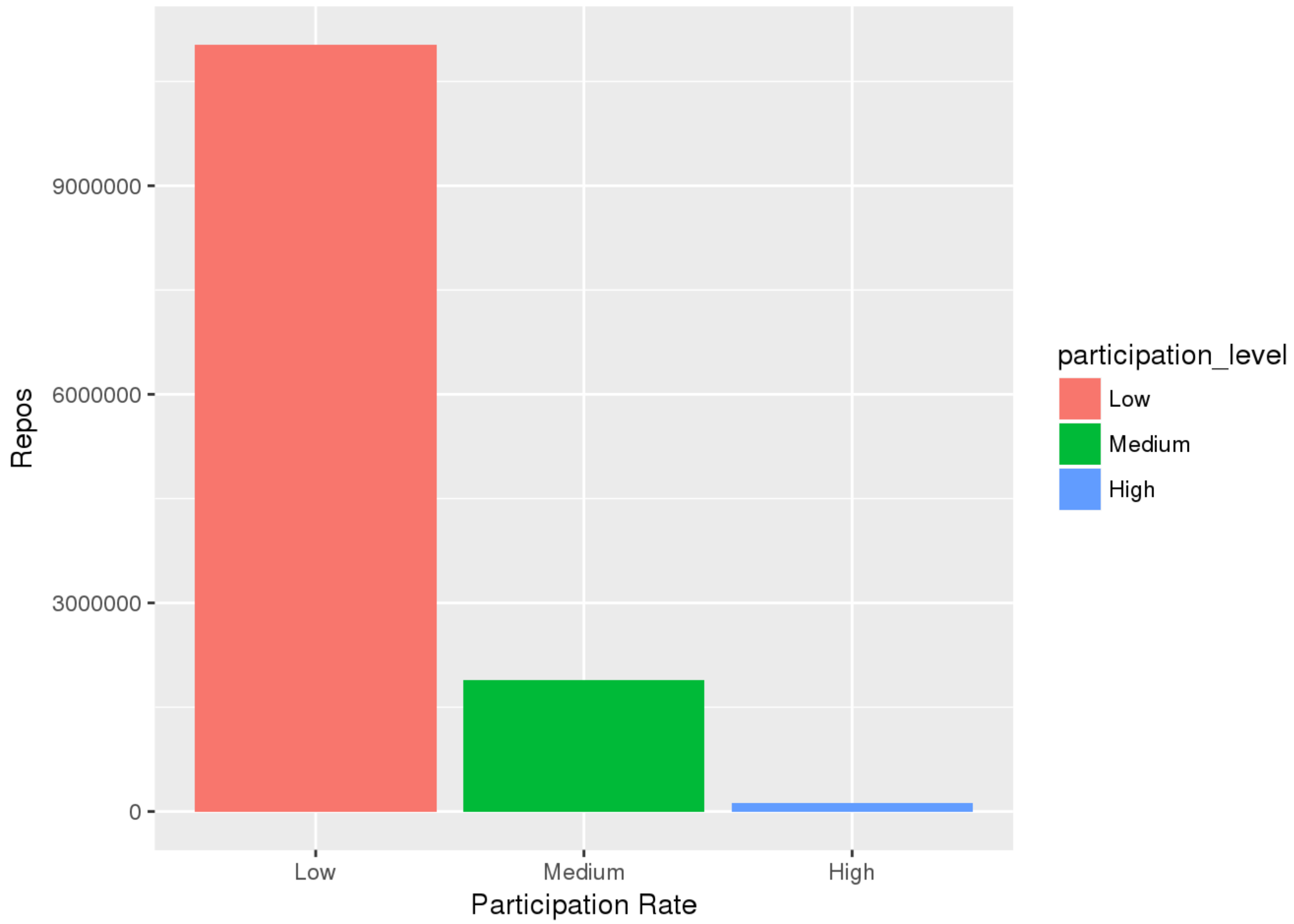
For example, for a total of 100 events: 1 unique actor = (100/1)/100 = 1 or 100% of events per actor 2 unique actors = (100/2)/100 = .5 or 50% of events per actor 5 unique actors = (100/5)/100 = .2 or 20% of events per actor 100 unique actors = (100/100)/100 = .01 1% of events per actor Lower value = higher participation level

I rounded to one decimal place to make the Participation Rate discrete rather than continous. Because of this rounding, repositories that fell into the rating of 0 and 1 had a significantly higher frequency than repositories that fell in to .1 through .5. To adjust for this, I defined 3 levels of Participation: High (0), Medium (.1-.5), and Low (1).

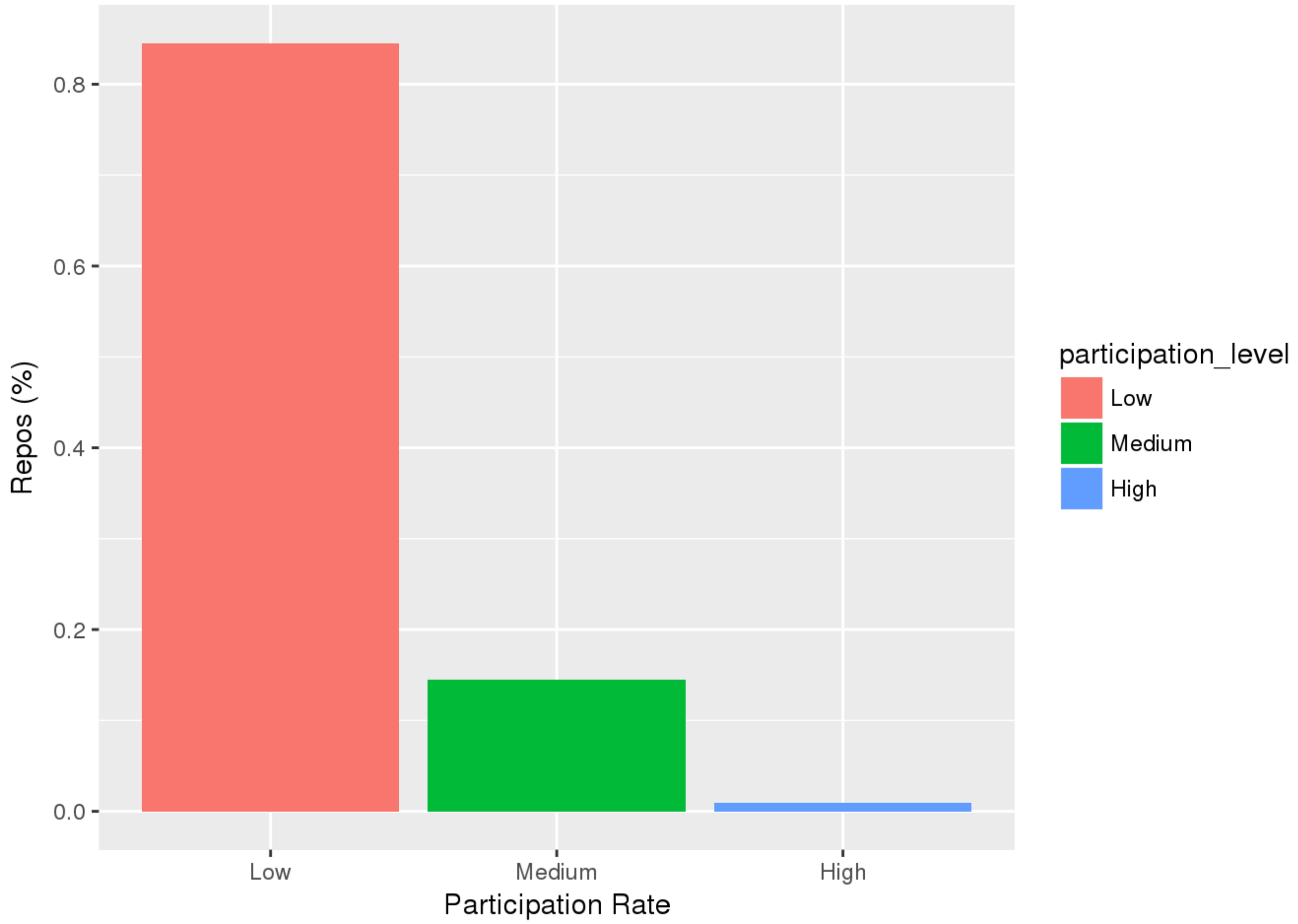
# What is the proportion of different participation levels in the repository population?

The majority of repositories in the population are Low participation, meaning they only have 1 unique actor generating events.

```
participation_rate_summary <- readRDS("participation_rate_summary.rds")

participation_rate_summary$participation_level <- factor(participation_rate_summary$participation_level,
  levels = unique(
    participation_rate_summary$participation_level[order(participation_rate_summary$num_repos, decreasing=TRUE)]))

ggplot(data = participation_rate_summary,
       aes(x=participation_level, y=num_repos, fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos") +
  scale_y_continuous(labels=function(n){format(n, scientific = FALSE)})
```



```
total_repos <- sum(participation_rate_summary$num_repos)
participation_rate_summary <- participation_rate_summary %>%
  mutate(repos_perc = num_repos/total_repos)

ggplot(data = participation_rate_summary,
       aes(x=participation_level, y=repos_perc, fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")
```



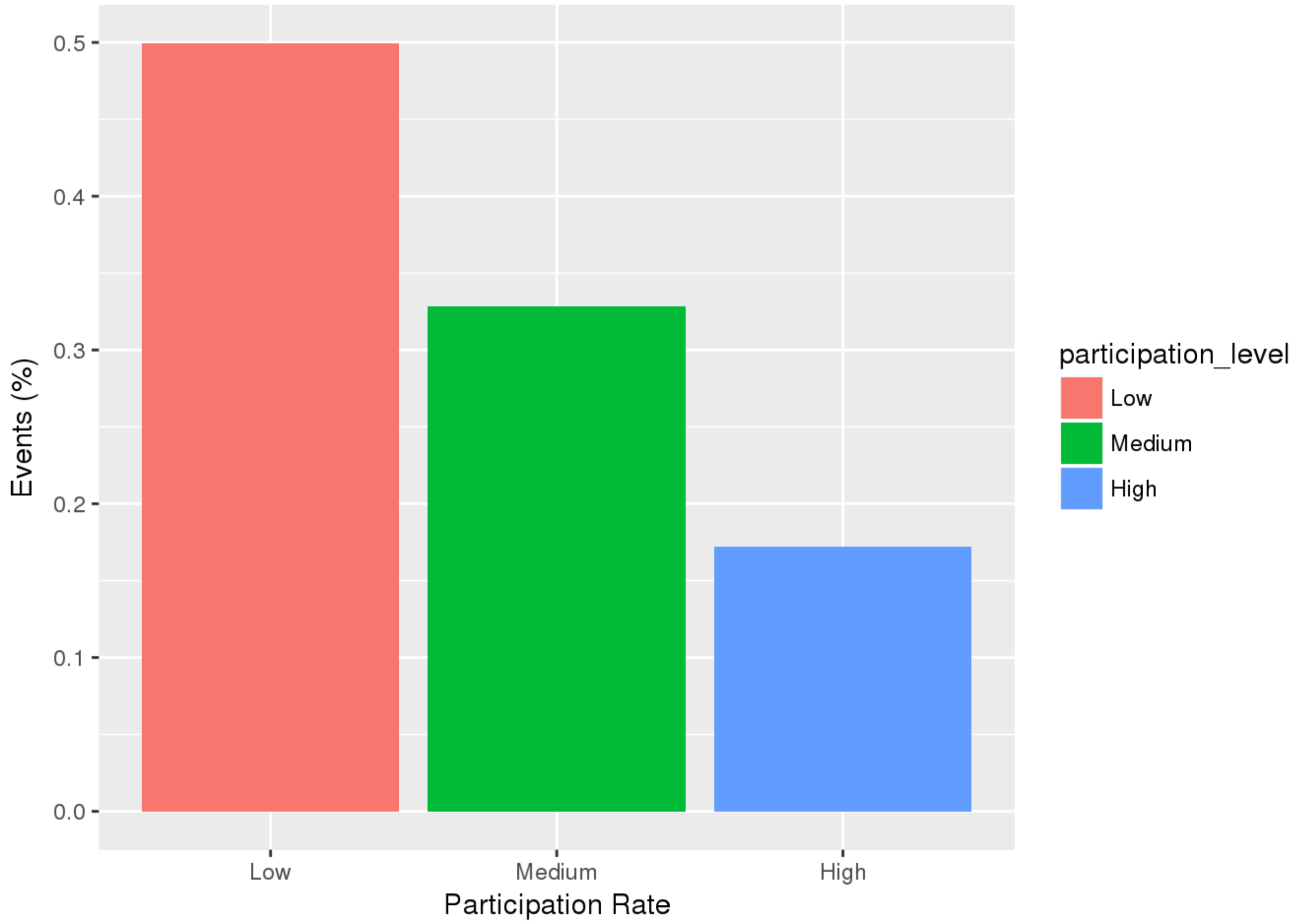
```
ggsave(filename="participation_repos_pct.png")
```

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

# What proportion of events is each participation rate responsible for?

Low participation repositories were responsible for almost half of the events. Medium participation repositories accounted for nearly 30% while High participation repositories accounted for under 20% of all event activity.

```
total_events <- sum(participation_rate_summary$num_events)

ggplot(data = participation_rate_summary,
       aes(x=participation_level, y=num_events/total_events,
fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Events (%)")
```



```
ggsave(filename="participation_events_pct.png")
```

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

# What proportion of repositories contributed to event activity in each participation level?

The next series of visualizations plot the relative event frequency among repositories in each participation level. The relative events frequency was calculated by counting the total number of events per repository and dividing that number by the total number of events in each participation level. A log transform has been applied to the event frequency to reduce the scale of the overall shape of the data.

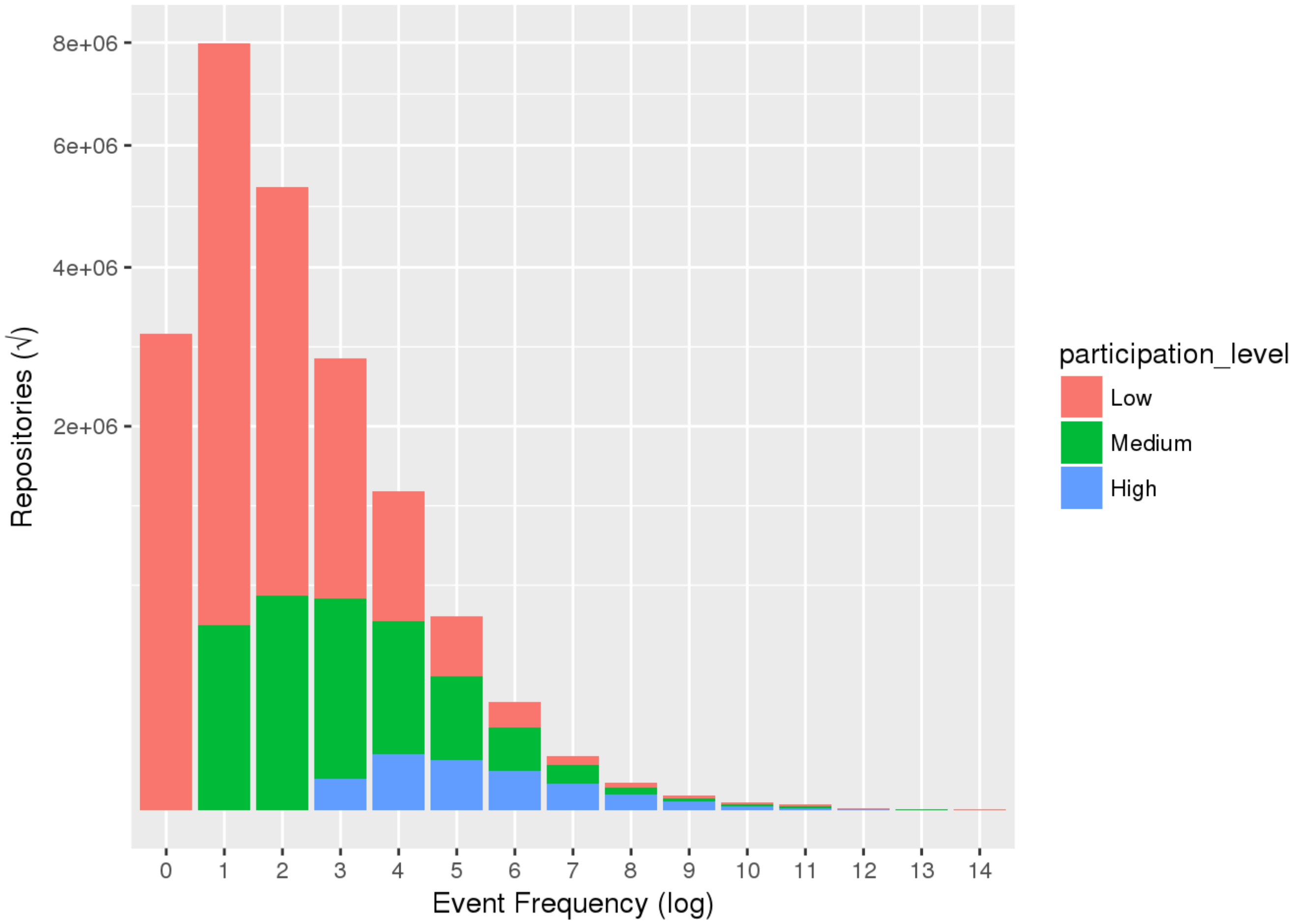
## All Participation Levels

As indicated in the previous plot, about half the events can be attributed to Low participation repositories. A square root transform has been applied to the repositories due to the difference in scale between each participation level.

The distribution is heavily right skewed, showing that the majority of repositories contributed to activity in the lower end of the events range. However, the minimum number of events represented by Medium and High activity repositories is much higher than the minimum number of events represented by the low activity repositories. Low participation repositories show the greatest spread overall with data points in both the lowest and highest ends.

```
participation_num_events_freq <- readRDS("participation_num_events_freq.rds")

participation_num_events_freq_sum <- participation_num_events_freq %>%
  group_by(participation_level, event_freq_log) %>%
  summarise(num_repos = sum(num_repos),
            event_freq_min = min(event_freq),
            event_freq_max = max(event_freq),
            event_freq_med = median(event_freq),
            event_freq_cnt = n())
```

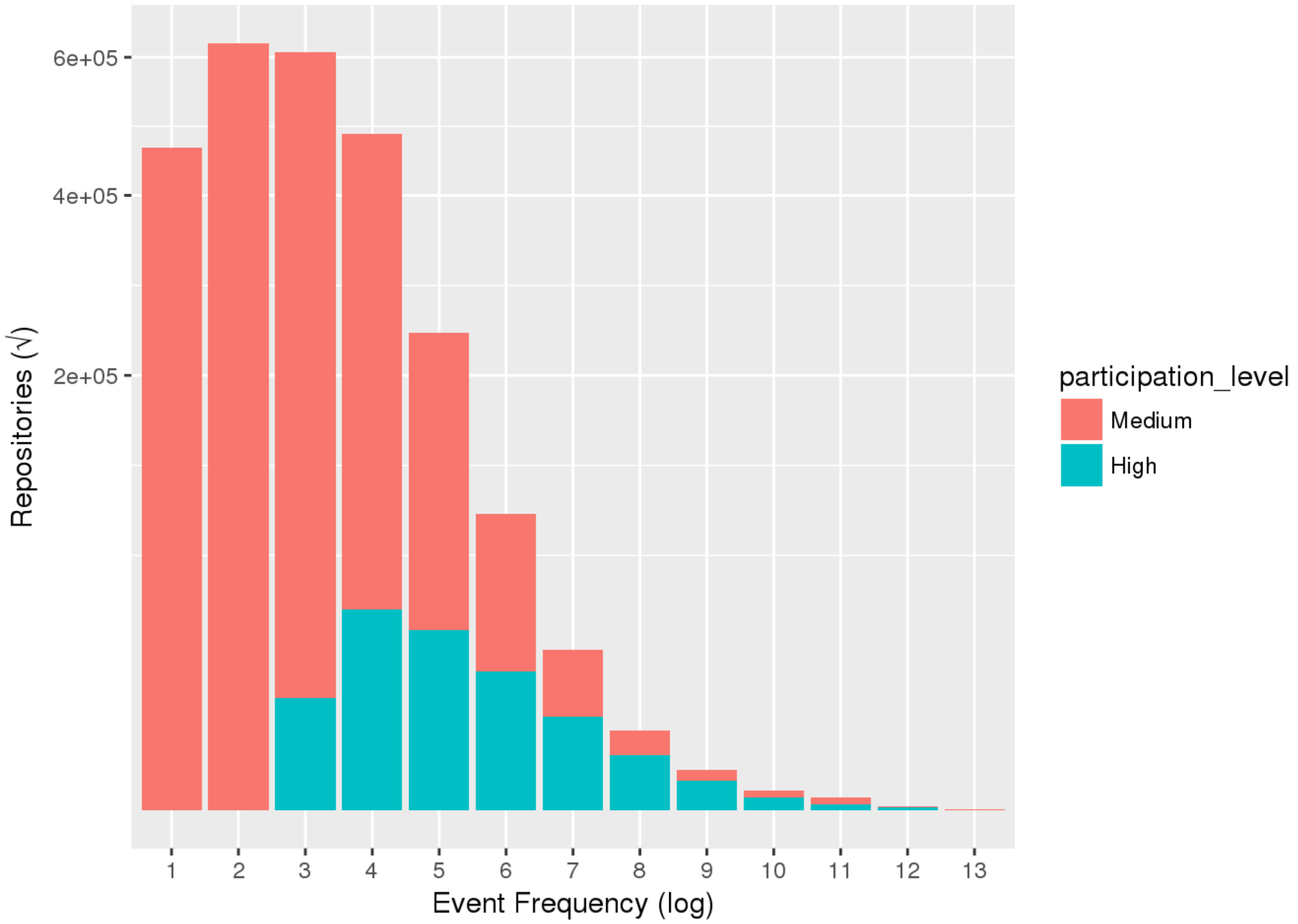
```
participation_num_events_freq_sum$participation_level <- factor(
  participation_num_events_freq_sum$participation_level,
  levels = unique(participation_num_events_freq_sum$participation_level[
    order(participation_num_events_freq_sum$num_repos, decreasing=TRUE)]))

ggplot(data = participation_num_events_freq_sum,
       aes(x=factor(event_freq_log), y=num_repos,
           fill = participation_level)) +
  geom_bar(stat="identity") +
  xlab("Event Frequency (log)") +
  ylab("Repositories (√)") +
  scale_y_continuous(trans = "sqrt")
```



```
ggsave(filename="participation_num_events_freq.png")
```

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

```
participation_num_events_freq_sum_mh <- participation_num_events_freq_sum %>%
  filter(participation_level != "Low")

  ggplot(data = participation_num_events_freq_sum_mh,
       aes(x=factor(event_freq_log), y=num_repos,
           fill = participation_level)) +
  geom_bar(stat="identity") +
  xlab("Event Frequency (log)") +
  ylab("Repositories (√)") +
  scale_y_continuous(trans = "sqrt")
```

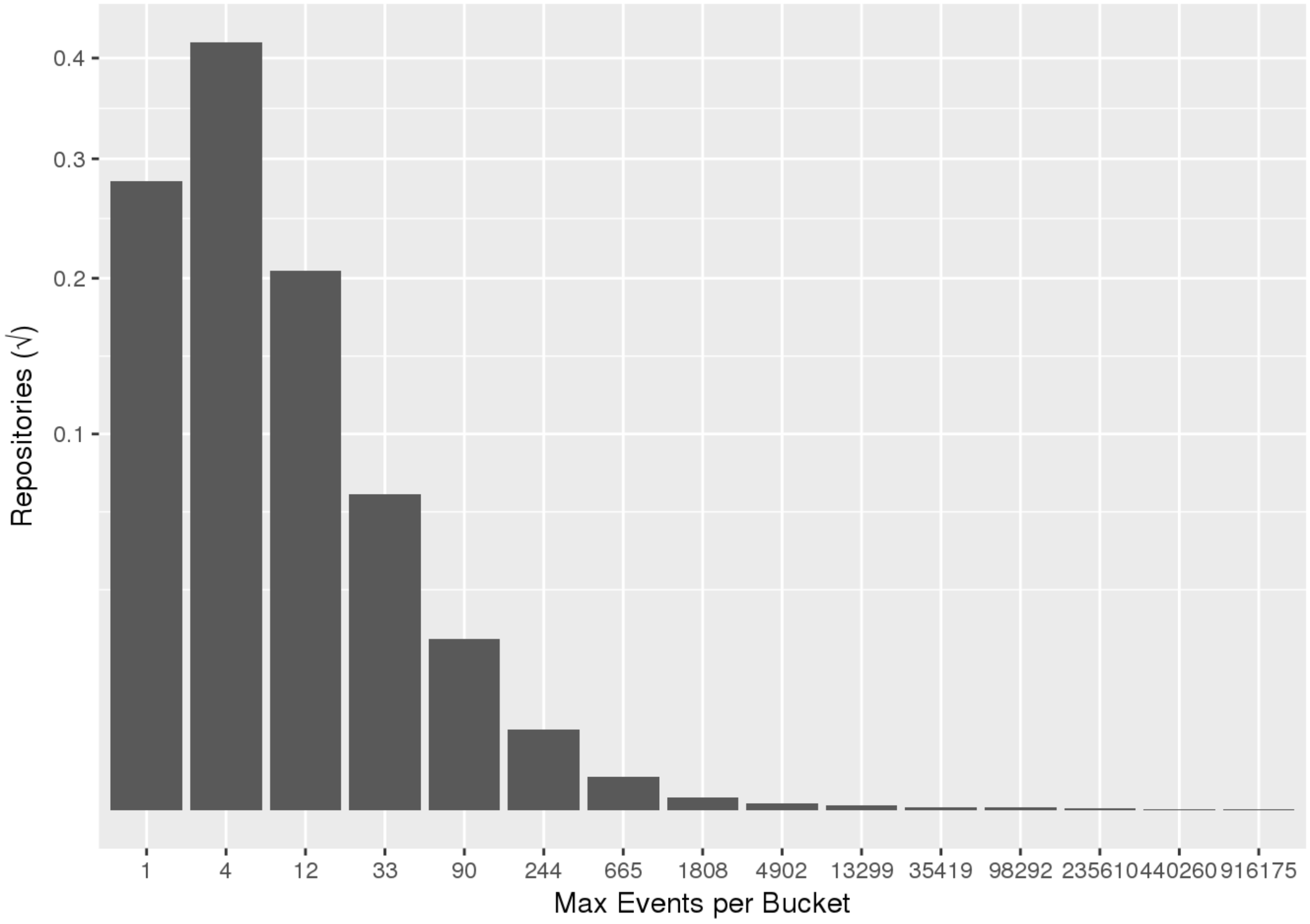


## Low Participation

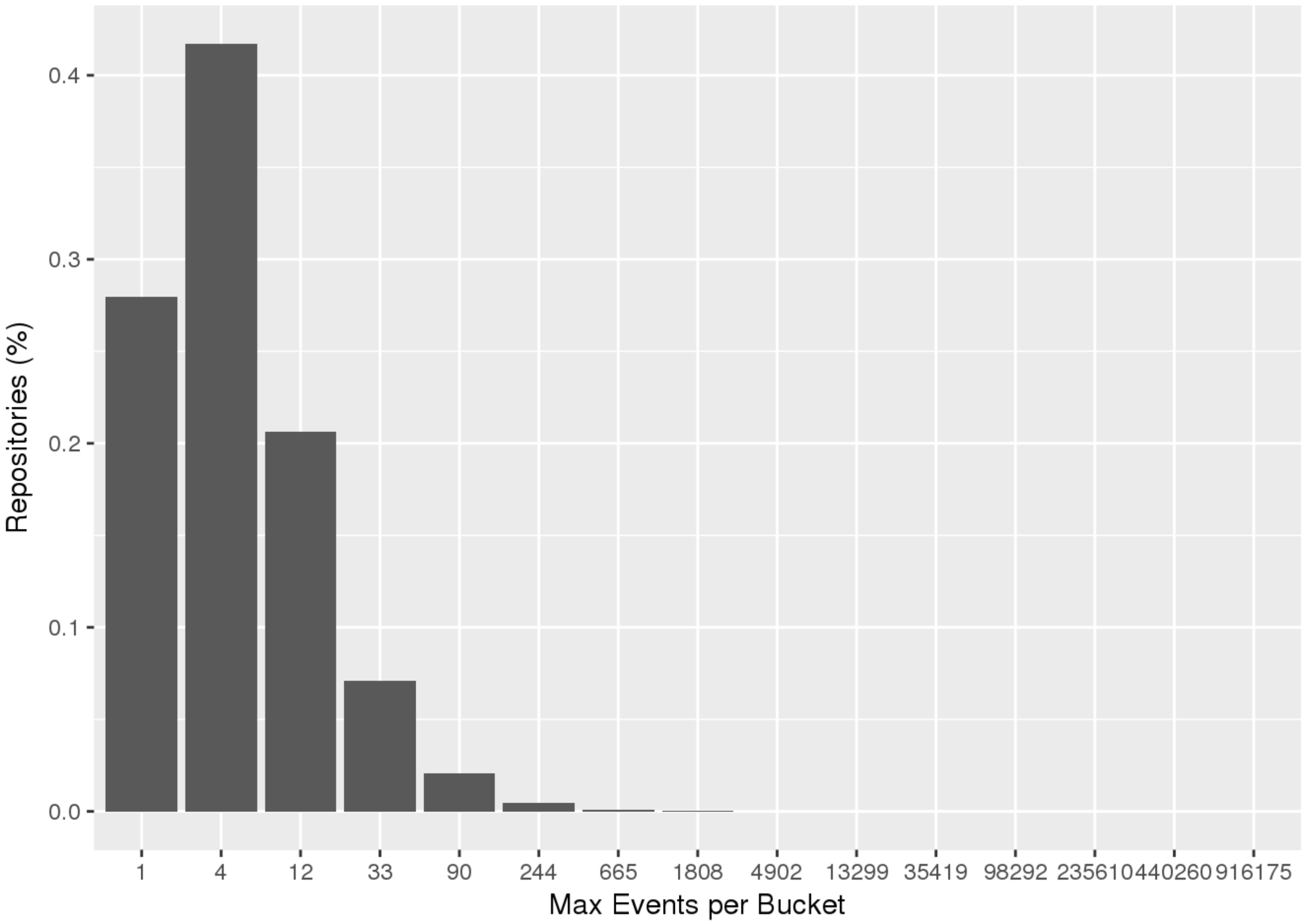
Around 70% of the repositories in this group accounted for 4 or less events per repository with a small number of repositories generating an incredibly large number of events for one contributor. This will be explored further down.

```
lp_summary <- participation_rate_summary %>% filter(participation_level == "Low")
low_participation_num_events_freq <- readRDS("low_participation_num_events_freq.rds")
low_participation_num_events_freq <- low_participation_num_events_freq %>%
  mutate(num_events_perc = (event_freq*num_repos)/total_events,
         num_lp_events_perc = (event_freq*num_repos)/lp_summary$num_events)
```

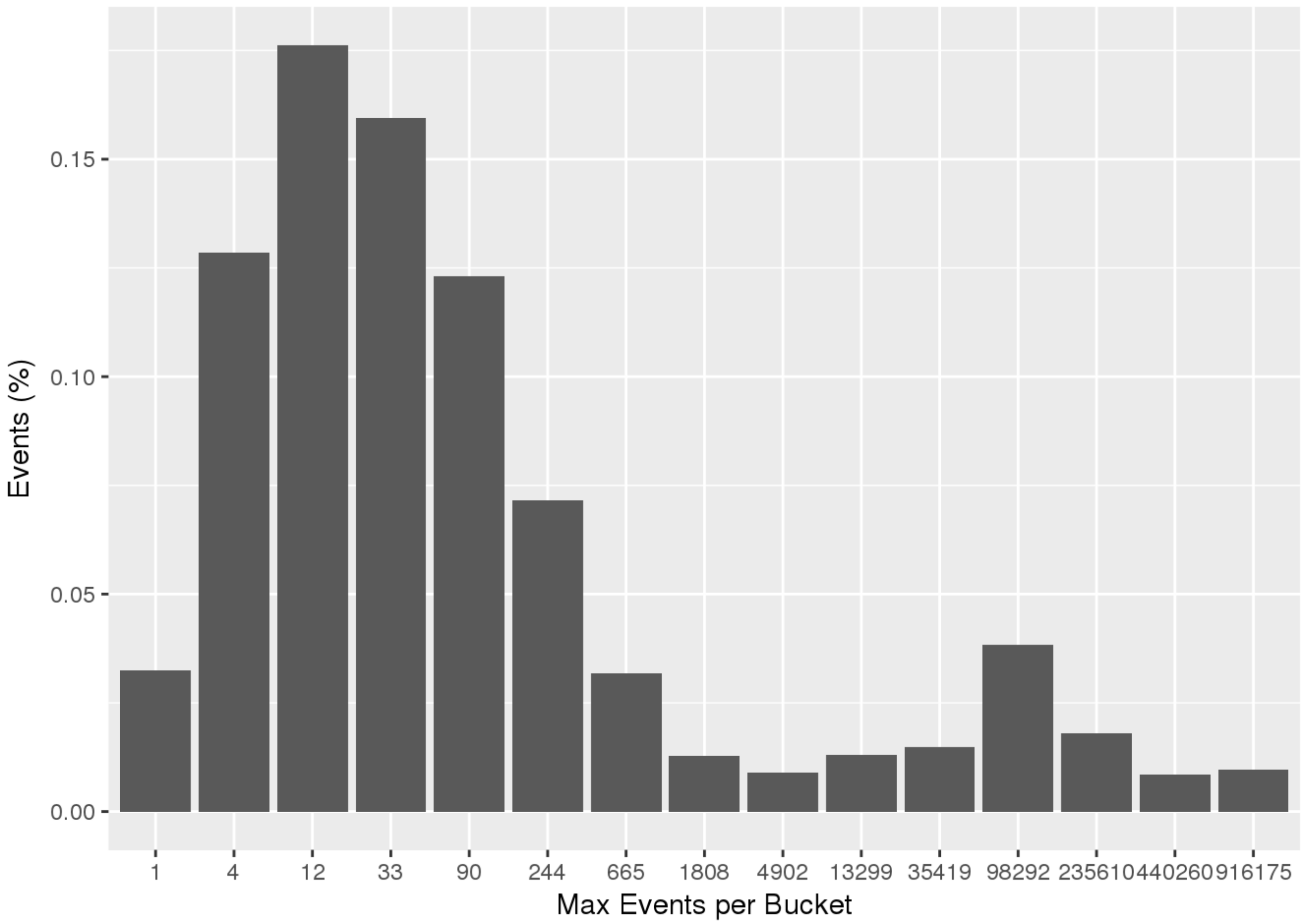
```
lp_events_freq_sum <- low_participation_num_events_freq %>%
  group_by(event_freq_log) %>%
  summarise(num_repos = sum(num_repos),
            event_freq_min = min(event_freq),
            event_freq_max = max(event_freq),
            event_freq_med = median(event_freq),
            event_freq_cnt = n(),
            num_lp_repos_perc = num_repos/lp_summary$num_repos,
            num_events_perc = sum(num_events_perc),
            num_lp_events_perc = sum(num_lp_events_perc))

ggplot(data = lp_events_freq_sum, aes(x=factor(event_freq_max), y=num_lp_repos_perc))
+
   geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Repositories (√)") +
  scale_y_continuous(trans = "sqrt")
```



```
ggplot(data = lp_events_freq_sum,
       aes(x=factor(event_freq_max), y=num_lp_repos_perc)) +
  geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Repositories (%)")
```



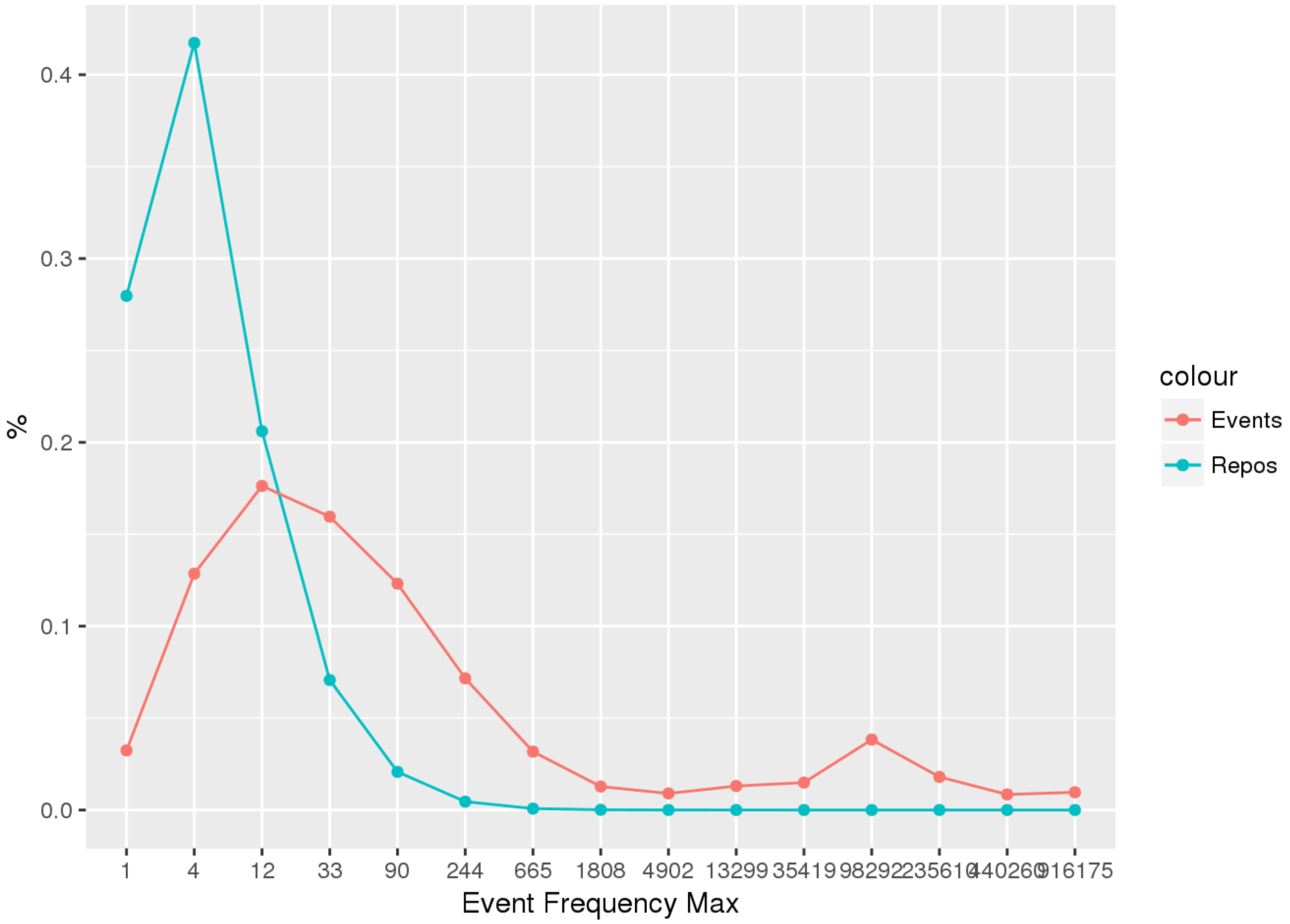
```
ggplot(data = lp_events_freq_sum,
       aes(x=factor(event_freq_max), y=num_lp_events_perc)) +
  geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Events (%)")
```



```
ggsave(filename="lp_events_freq_sum.png")
```

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

```
ggplot(data = lp_events_freq_sum,
       aes(x=factor(event_freq_max))) +
  geom_point(stat="identity", aes(y=num_lp_repos_perc, color="Repos")) +
  geom_line(stat="identity", aes(y=num_lp_repos_perc, color="Repos", group=1)) +
  geom_point(stat="identity", aes(y=num_lp_events_perc, color="Events")) +
  geom_line(stat="identity", aes(y=num_lp_events_perc, color="Events", group=1)) +
  xlab("Event Frequency Max") +
  ylab("%")
```



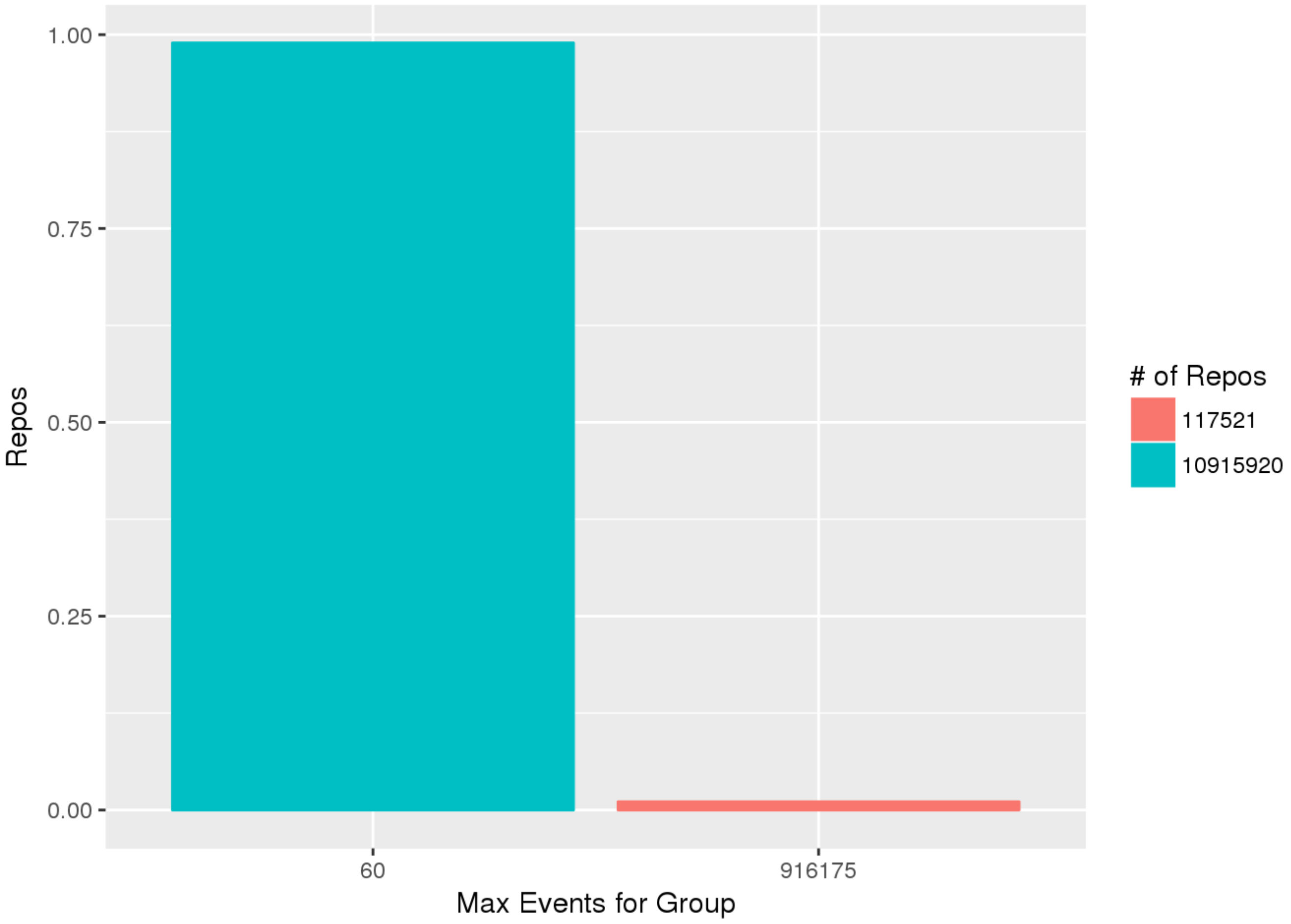
```
ggsave(filename="lp_events_freq_sum_vs_repos.png")
```

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

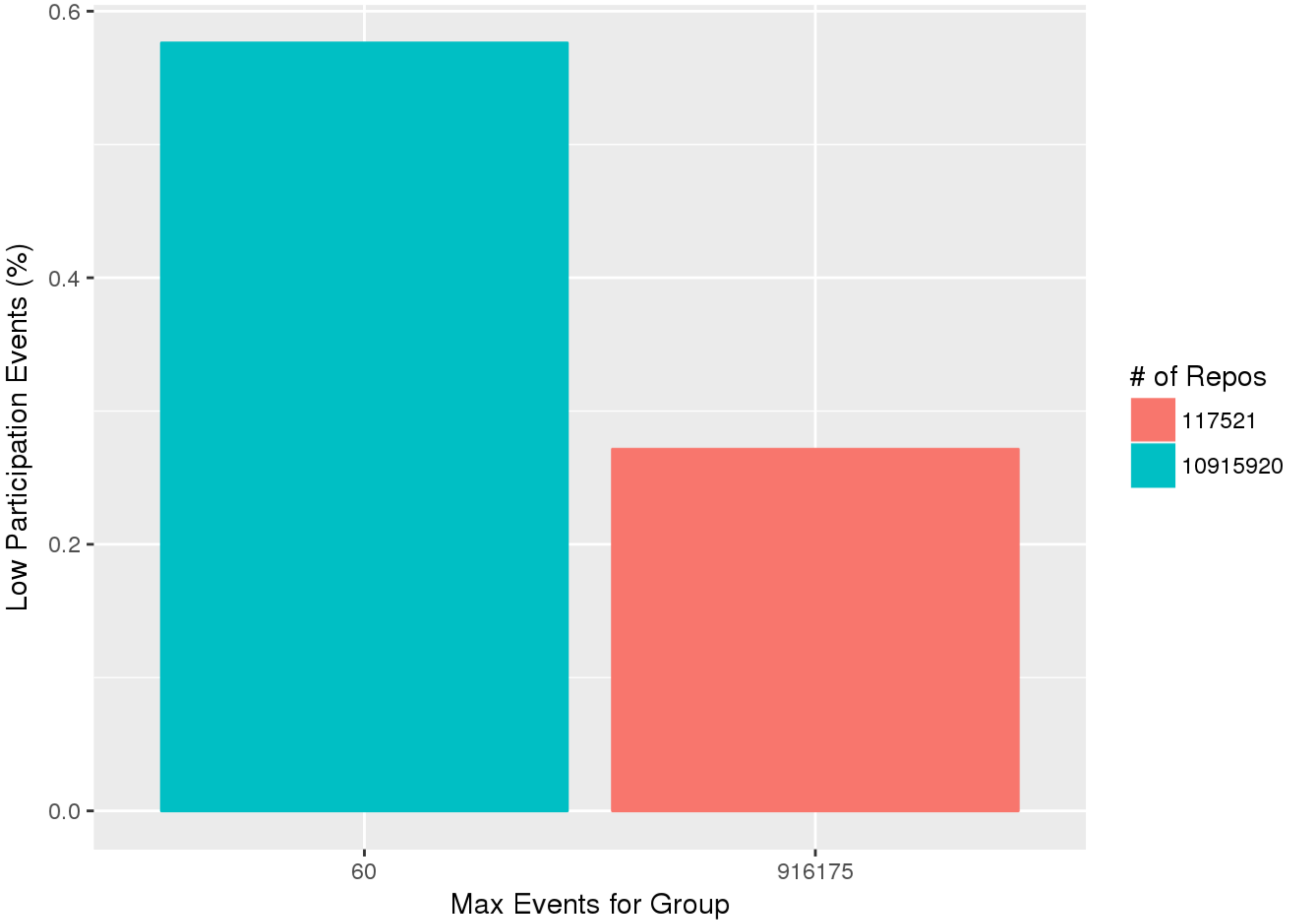
The majority of Low participation repositories (10,934,829) had 60 or less events per repository out of 11,033,441, or about 99% of the population studied. Over 6 months, 60 events would be a maximum of 10 events per month. This represents just under 60% of all Low participation event activity.

Almost 30% of all event activity came from repositories with 1 actor and 60 or less total events for the 6 month time period.

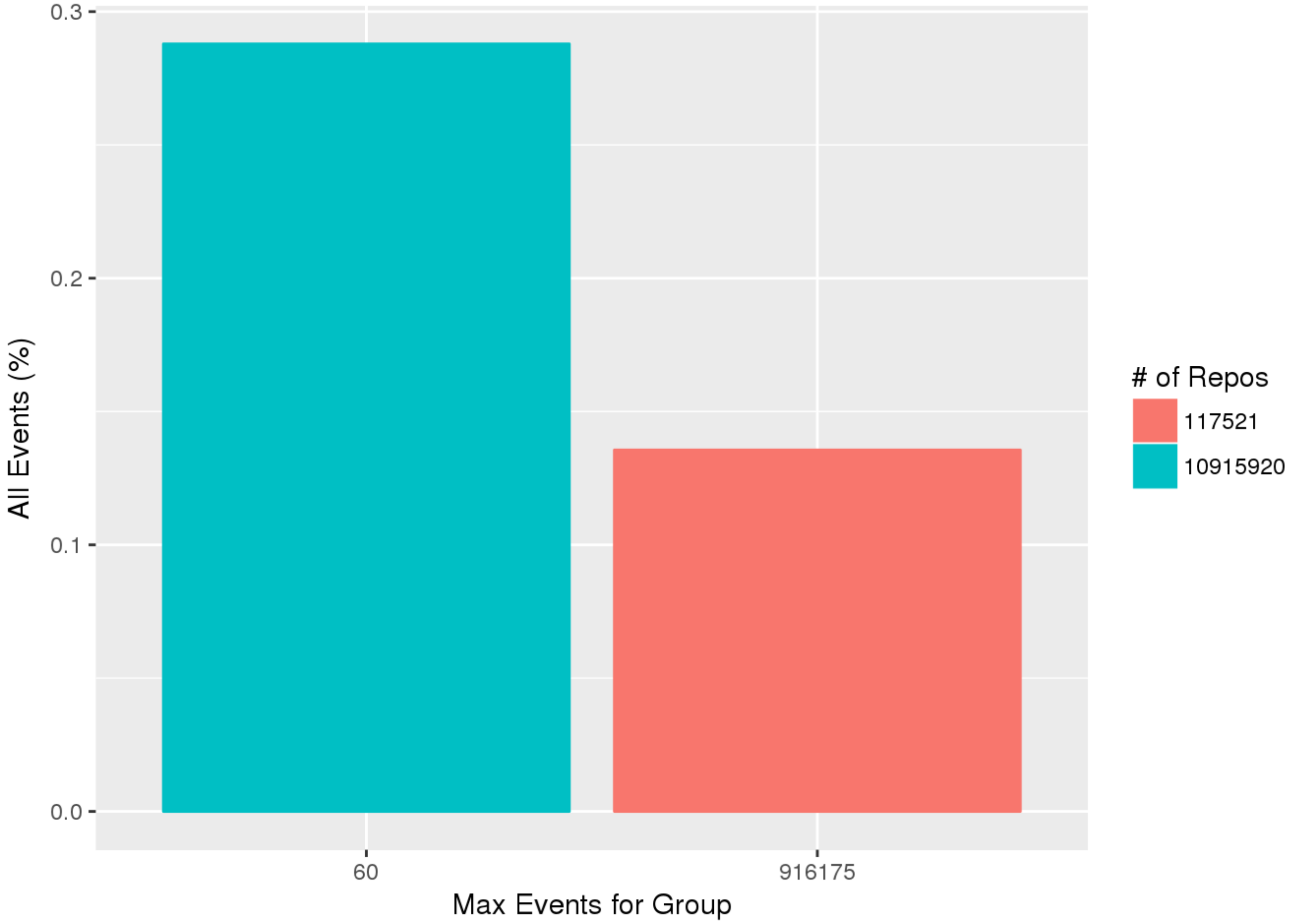
```
lp_events_freq_split <- low_participation_num_events_freq %>%
  mutate(event_freq_group = ifelse(event_freq > 60, 2, 1)) %>%
  group_by(event_freq_group) %>%
  summarise(num_repos = sum(num_repos),
            event_freq_min = min(event_freq),
            event_freq_max = max(event_freq),
            event_freq_med = median(event_freq),
            event_freq_mean = round(mean(event_freq)),
            event_freq_cnt = n(),
            num_events_perc = sum(num_events_perc),
            num_lp_events_perc = sum(num_lp_events_perc))

# TODO this might be duplicated
lp_events_freq_split <- lp_events_freq_split %>%
  mutate(num_lp_repos_perc = num_repos/lp_summary$num_repos)

ggplot(data = lp_events_freq_split,
       aes(x=factor(event_freq_max),
           y=num_lp_repos_perc,
           color=factor(num_repos), fill = factor(num_repos))) +
  geom_bar(stat="identity") +
  labs(x = "Max Events for Group", y = "Repos", fill = "# of Repos", color = "# of Rep
os")
```



```
ggplot(data = lp_events_freq_split,
       aes(x=factor(event_freq_max),
           y=num_lp_events_perc,
           color=factor(num_repos), fill = factor(num_repos))) +
  geom_bar(stat="identity") +
  labs(x = "Max Events for Group", y = "Low Participation Events (%)", fill = "# of Re
pos", color = "# of Repos")
```



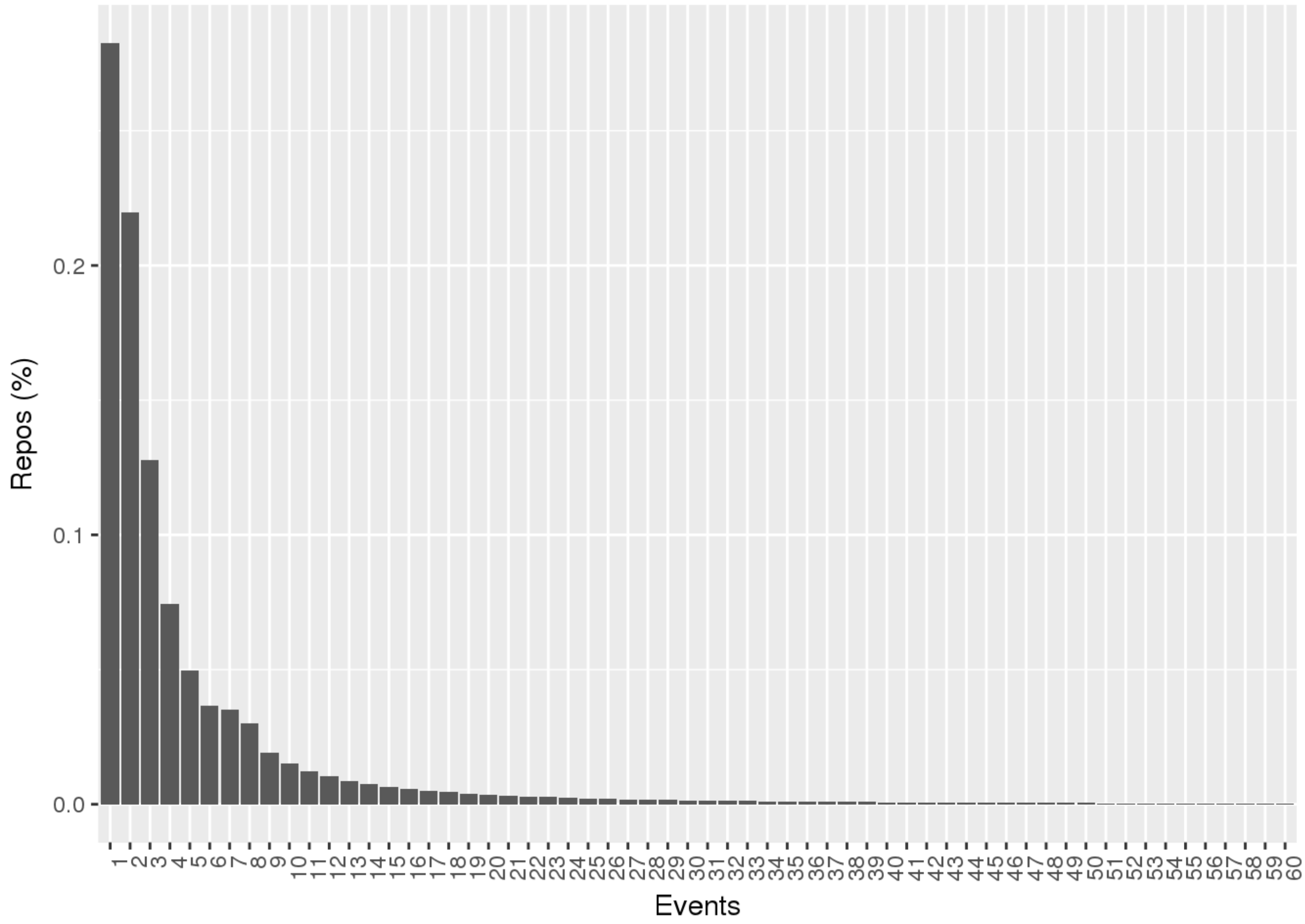
```
ggplot(data = lp_events_freq_split,
       aes(x=factor(event_freq_max),
           y=num_events_perc,
           color=factor(num_repos), fill = factor(num_repos))) +
  geom_bar(stat="identity") +
  labs(x = "Max Events for Group", y = "All Events (%)", fill = "# of Repos", color =
"# of Repos")
```



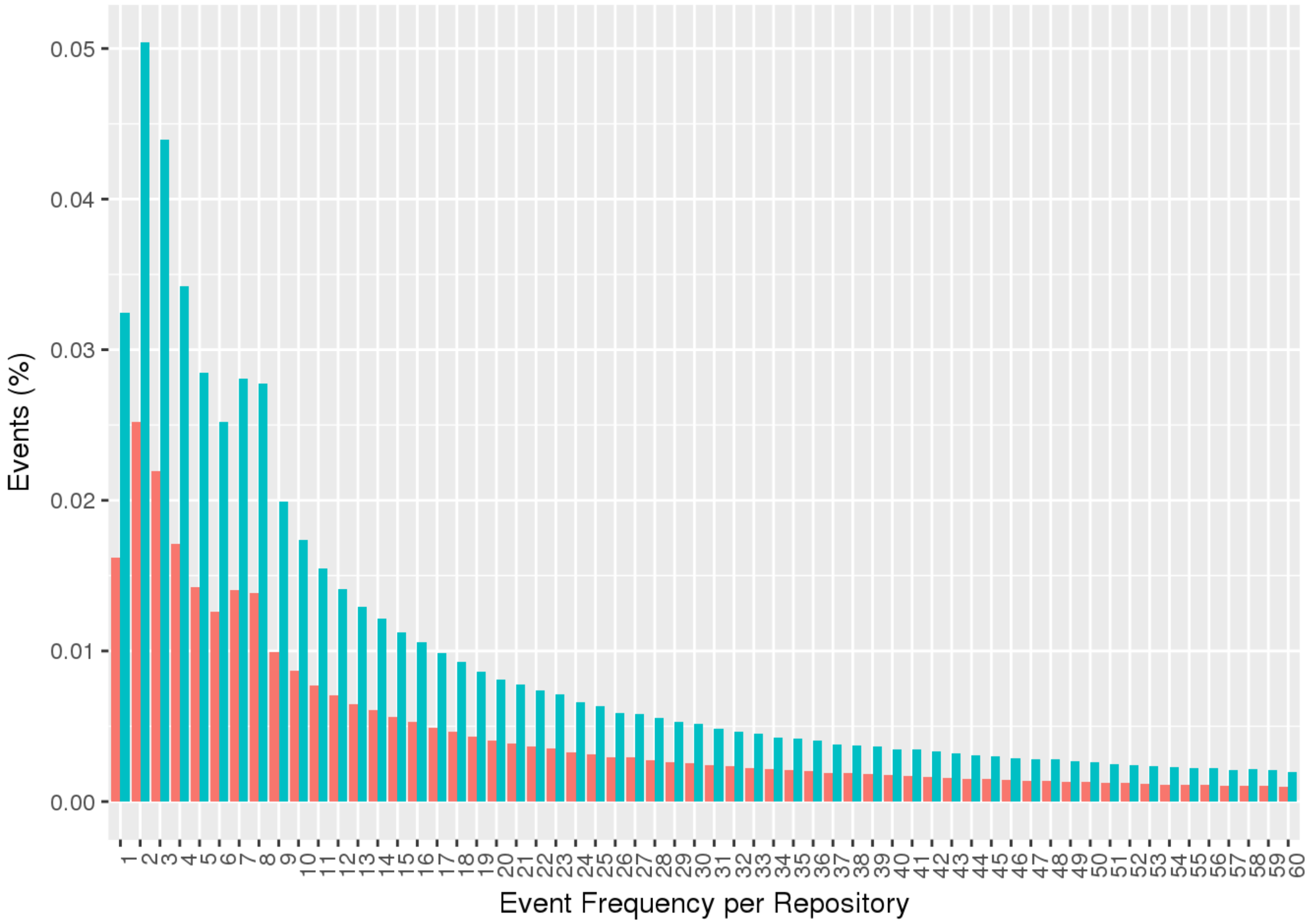
Below is a plot of two percentages, the shorter bar is the percent of events overall and the taller bar is percent of events within the Low participation events.

28% of repositories that had 60 or less events had only one event per repository. This represents just over 3% of all event activity during the 6 month period.

```
lp_events_freq_group1 <- low_participation_num_events_freq %>%
  filter(event_freq < 61)

lp_g1_total_repos <- sum(lp_events_freq_group1$num_repos)

# TODO this might be duplicated
lp_events_freq_group1 <- lp_events_freq_group1 %>%
  mutate(num_lp_repos_perc = num_repos/lp_g1_total_repos)

ggplot(data = lp_events_freq_group1,
       aes(x=factor(event_freq),
           y=num_lp_repos_perc)) +
  geom_bar(stat="identity") +
  labs(x = "Events", y = "Repos (%)") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
lp_events_freq_group1_long <- lp_events_freq_group1 %>%
  select(event_freq, num_events_perc, num_lp_events_perc) %>%
  melt(id.vars = "event_freq")

ggplot(data = lp_events_freq_group1_long, aes(x = factor(event_freq), y = value, fill
= variable)) +
  geom_bar(stat="identity", position="dodge") +
  labs(x = "Event Frequency per Repository", y = "Events (%)") +
  theme(legend.position="none") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

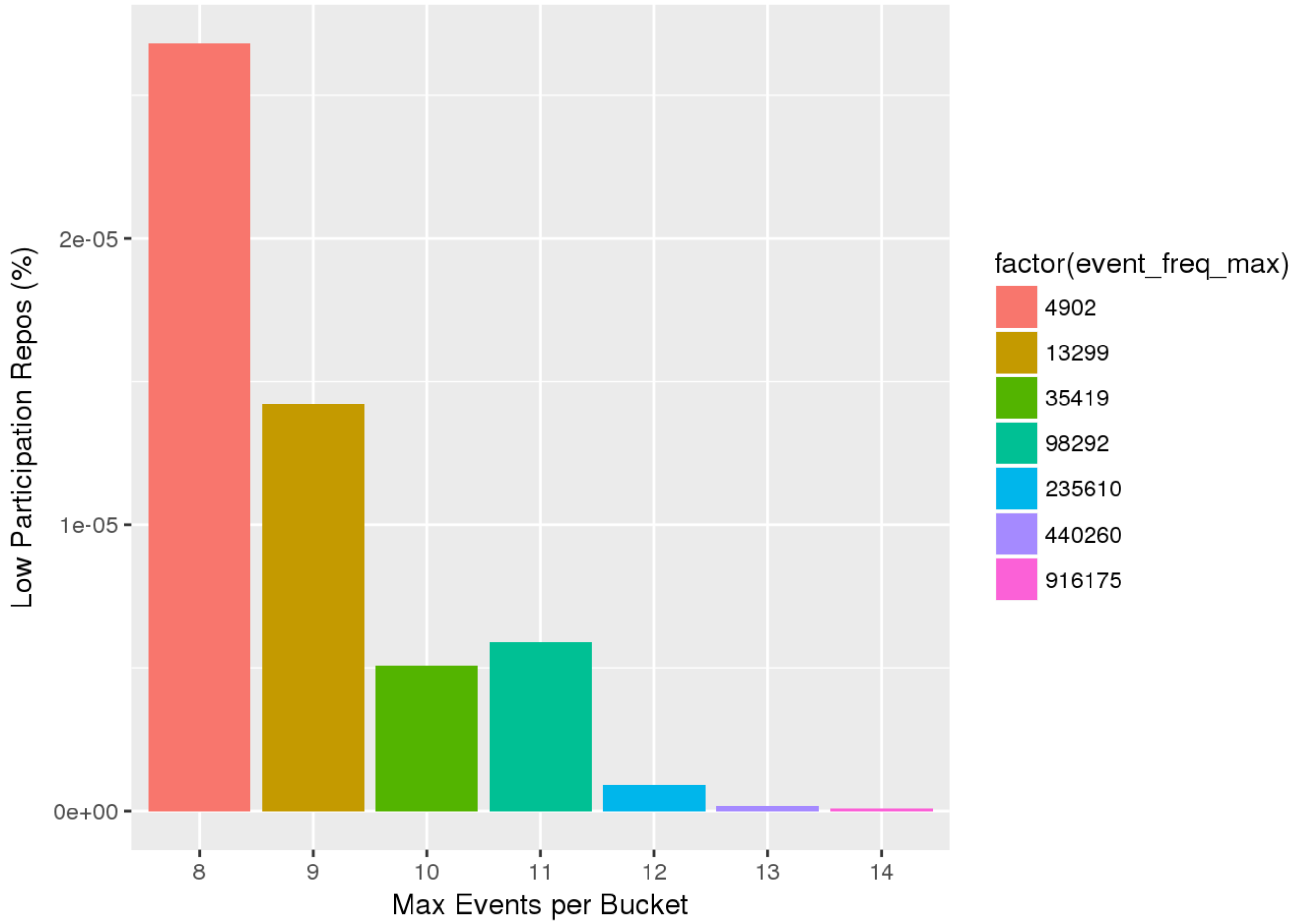


On the high end, 587 out of 11,033,441 or about .005% of the Low participation repositories had more than 1500 events over the 6 month period. For context, there are 213 days represented in the population, so that's an average of 7 events per day for just 1 person. On the very highest end, one repository accounted for 916,175 events (about 4,300 events per day) while 12 others accounted for over 100,000 events each (around 984 events per day)!

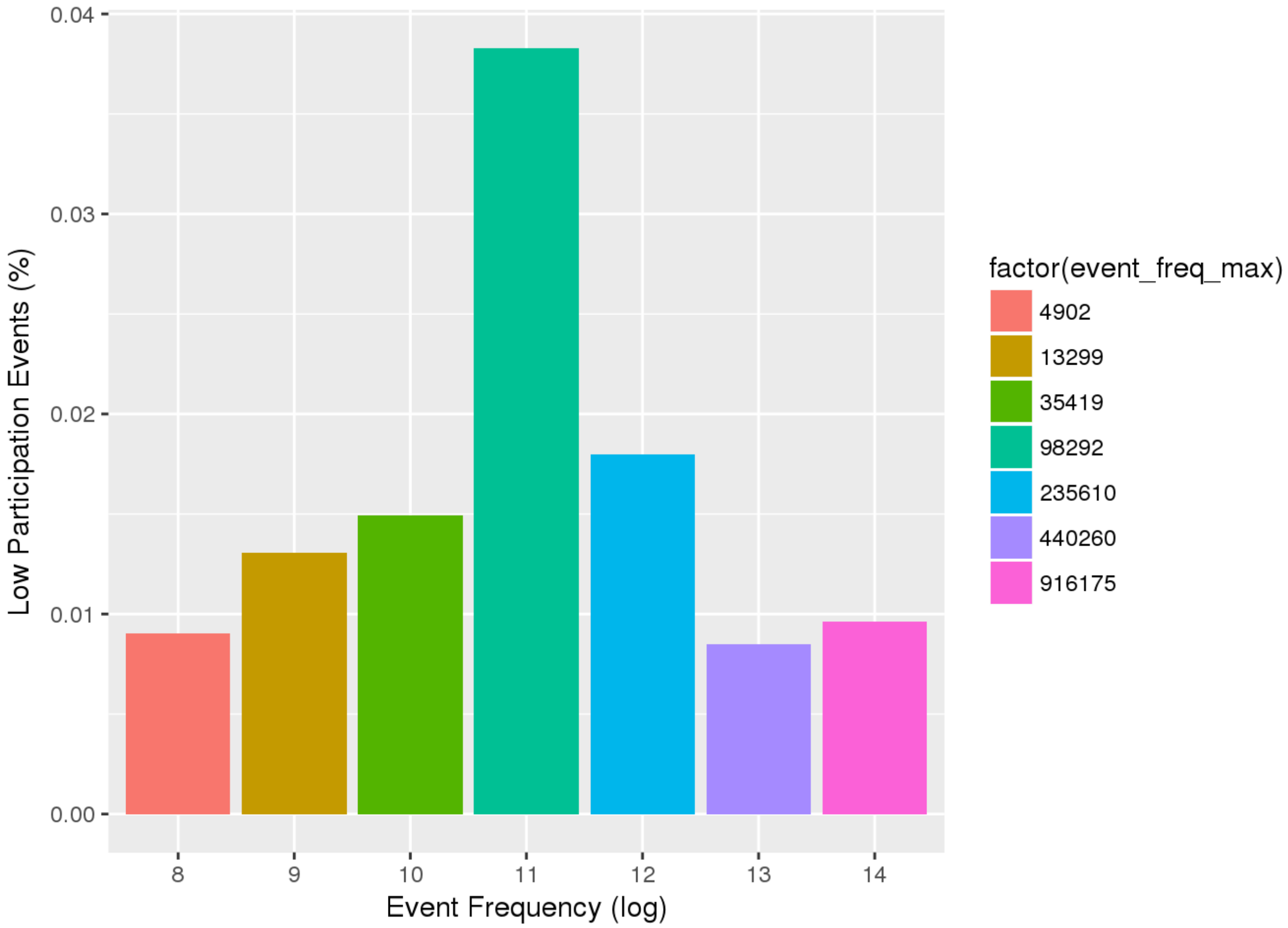
```
lp_events_freq_group2 <- lp_events_freq_sum %>%
  filter(event_freq_min > 1500)

lp_group2_events_repos_sum <- sum(lp_events_freq_group2$num_repos)
lp_group2_events_perc_sum <- sum(lp_events_freq_group2$num_events_perc)
lp_group2_lp_events_perc_sum <- sum(lp_events_freq_group2$num_lp_events_perc)
data.frame(repos=lp_group2_events_repos_sum,
      repos_pct=lp_group2_events_repos_sum/lp_summary$num_repos,
      events_pct= lp_group2_events_perc_sum,
      lp_events_pct= lp_group2_lp_events_perc_sum)
```

```
##   repos   repos_pct events_pct lp_events_pct
## 1   587 5.32019e-05 0.05565159     0.1113807
```

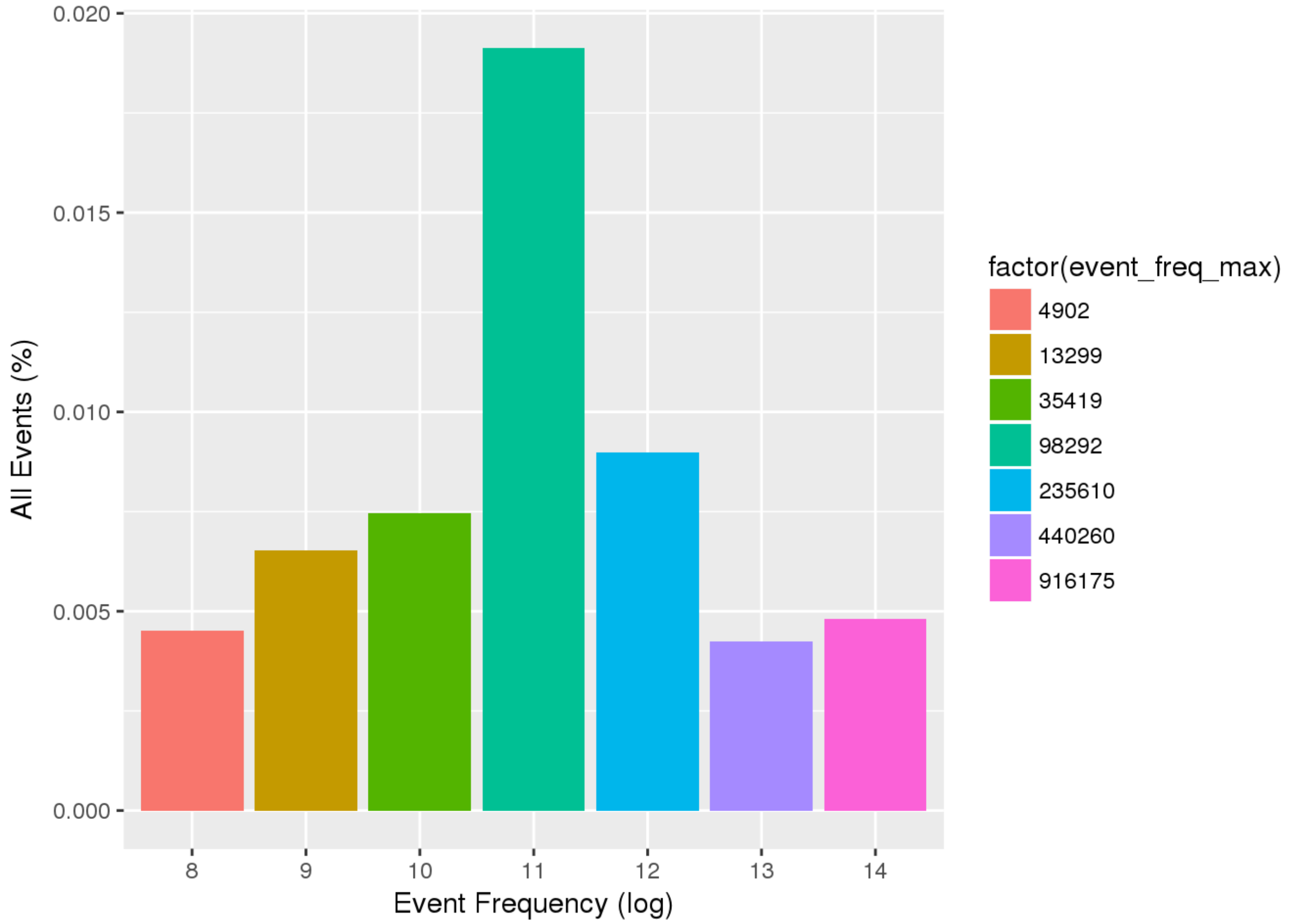
```
ggplot(data = lp_events_freq_group2,
       aes(x=factor(event_freq_log), y=num_lp_repos_perc,
fill=factor(event_freq_max))) +
  geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Low Participation Repos (%)")
```



```
ggplot(data = lp_events_freq_group2,
       aes(x=factor(event_freq_log), y=num_lp_events_perc,
fill=factor(event_freq_max))) +
  geom_bar(stat="identity") +
  xlab("Event Frequency (log)") +
  ylab("Low Participation Events (%)")
```



```
ggplot(data = lp_events_freq_group2,
       aes(x=factor(event_freq_log), y=num_events_perc, fill=factor(event_freq_max)))
+
  geom_bar(stat="identity") +
  xlab("Event Frequency (log)") +
  ylab("All Events (%)")
```



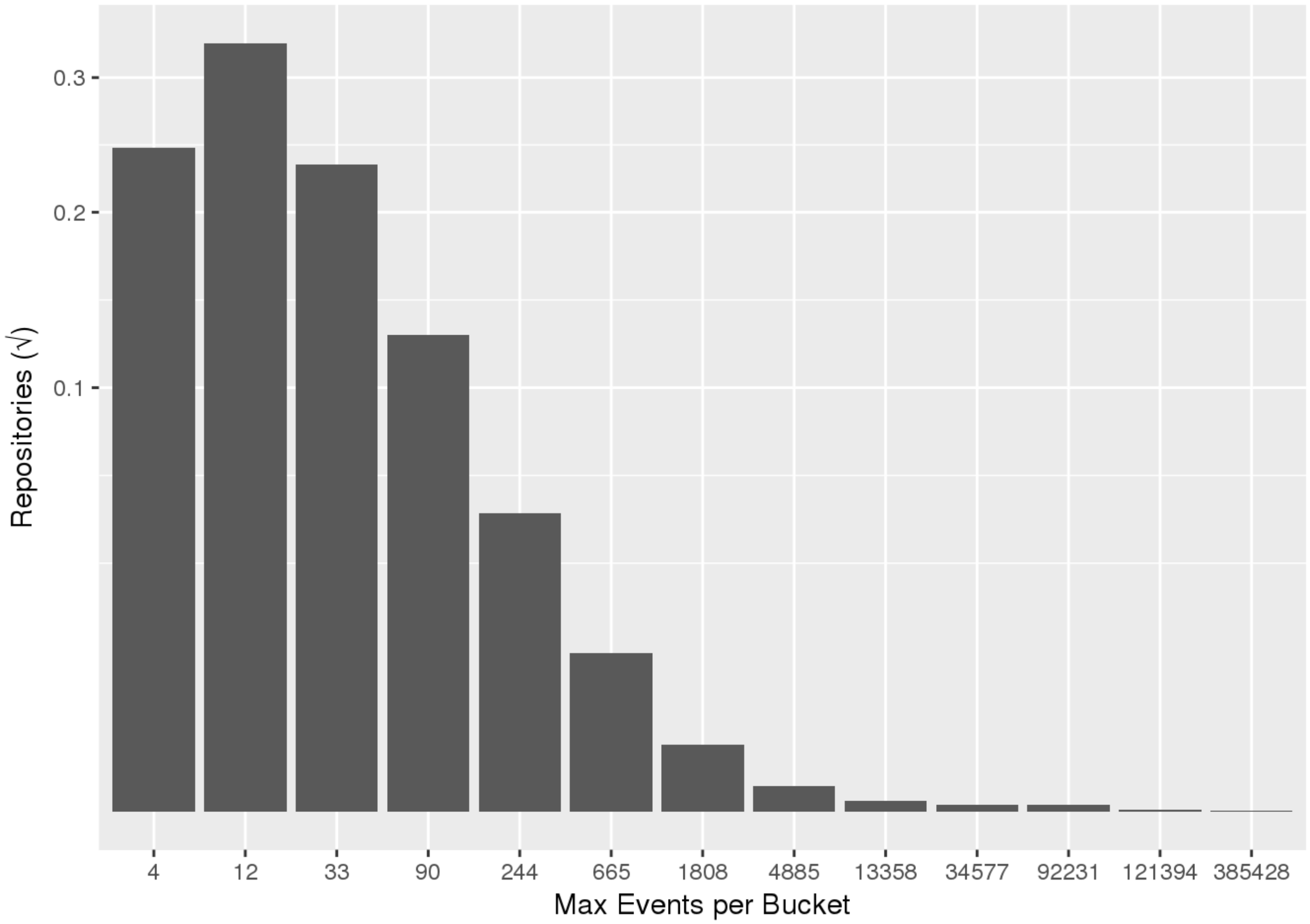
## Medium Participation

For Medium Participation repositories, about 75% of the repositories had between 2 and 33 events in the 6 month period.

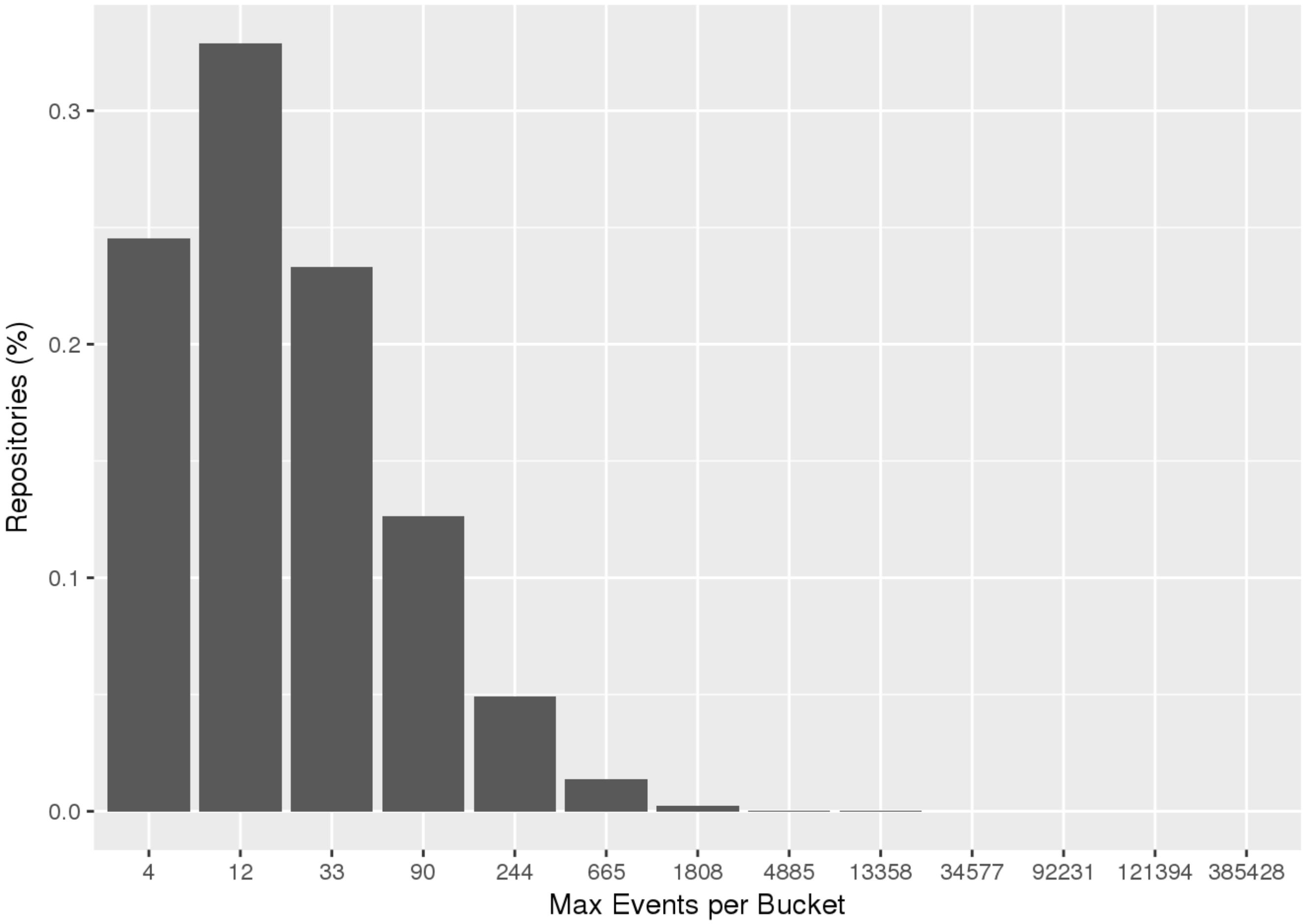
```
mp_summary <- participation_rate_summary %>% filter(participation_level == "Medium")
```

```
med_participation_num_events_freq <- readRDS("med_participation_num_events_freq.rds")
med_participation_num_events_freq <- med_participation_num_events_freq %>%
  mutate(num_events_perc = (event_freq*num_repos)/total_events,
         num_mp_events_perc = (event_freq*num_repos)/mp_summary$num_events)
```

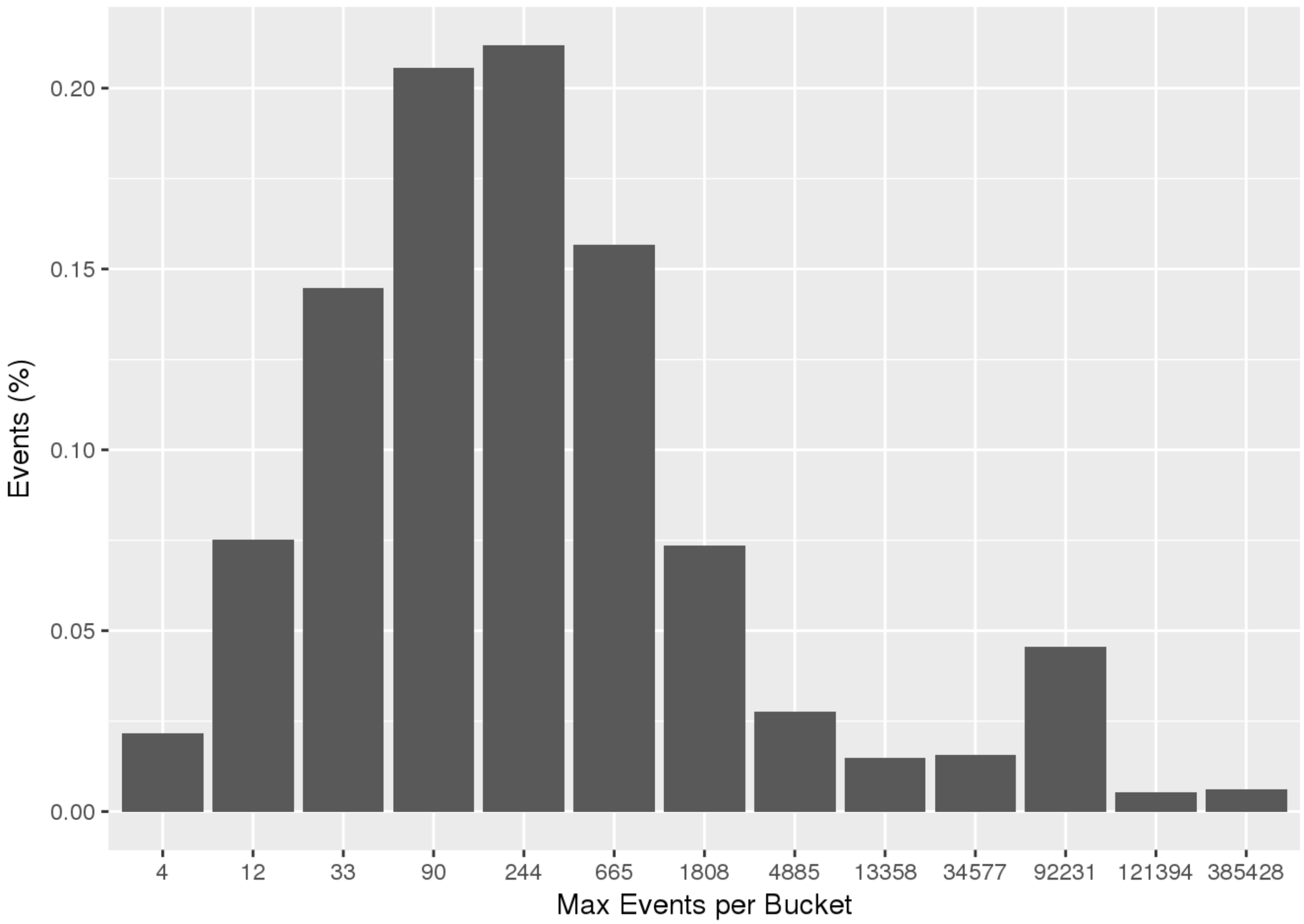
```
mp_events_freq_sum <- med_participation_num_events_freq %>%
  group_by(event_freq_log) %>%
  summarise(num_repos = sum(num_repos),
            event_freq_min = min(event_freq),
            event_freq_max = max(event_freq),
            event_freq_med = median(event_freq),
            event_freq_cnt = n(),
            num_mp_repos_perc = num_repos/mp_summary$num_repos,
            num_events_perc = sum(num_events_perc),
            num_mp_events_perc = sum(num_mp_events_perc))

ggplot(data = mp_events_freq_sum, aes(x=factor(event_freq_max), y=num_mp_repos_perc))
+
   geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Repositories (√)") +
  scale_y_continuous(trans = "sqrt")
```



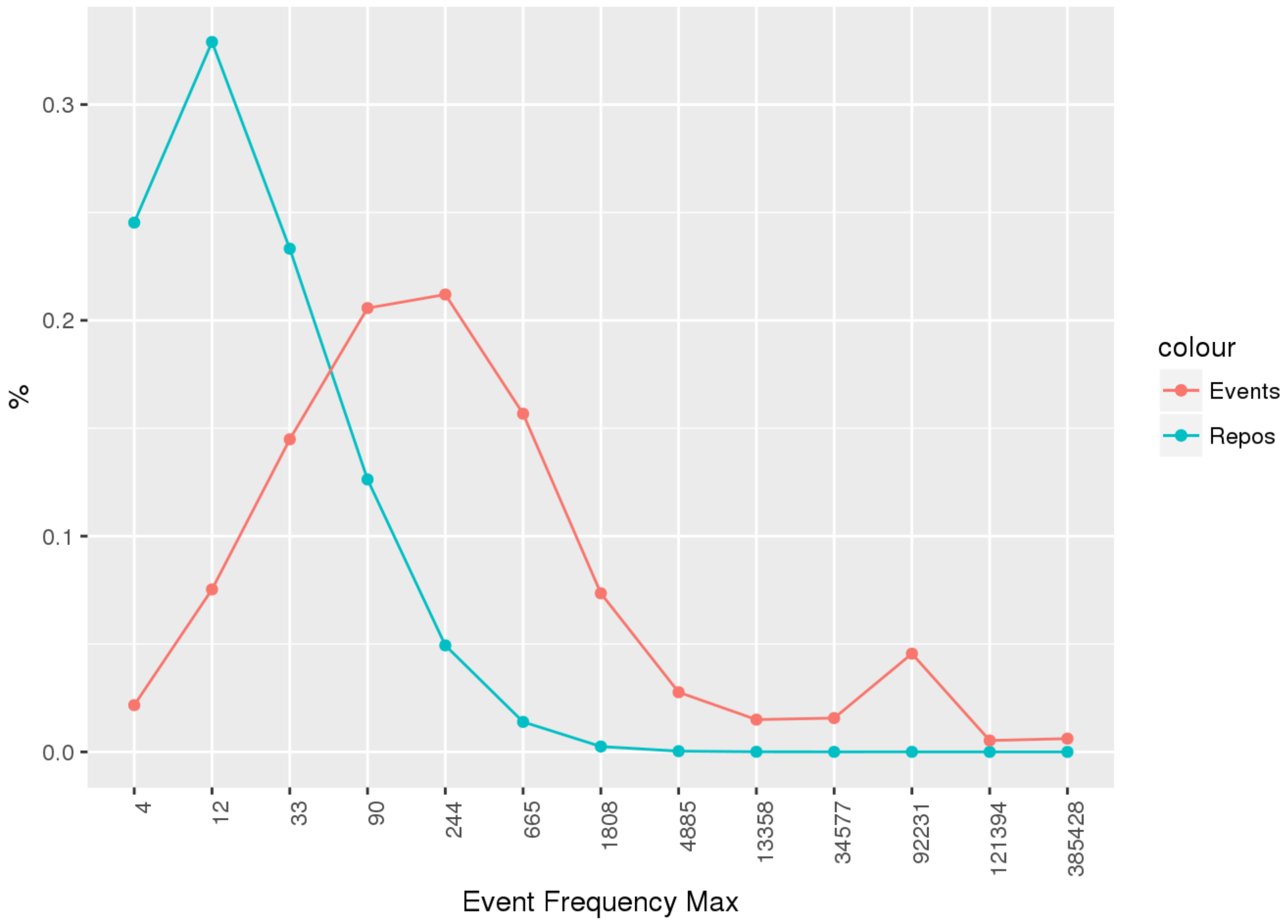
```
ggplot(data = mp_events_freq_sum,
       aes(x=factor(event_freq_max), y=num_mp_repos_perc)) +
  geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Repositories (%)")
```



```
ggplot(data = mp_events_freq_sum,
       aes(x=factor(event_freq_max), y=num_mp_events_perc)) +
  geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Events (%)")
```



```
ggplot(data = mp_events_freq_sum,
       aes(x=factor(event_freq_max))) +
  geom_point(stat="identity", aes(y=num_mp_repos_perc, color="Repos")) +
  geom_line(stat="identity", aes(y=num_mp_repos_perc, color="Repos", group=1)) +
  geom_point(stat="identity", aes(y=num_mp_events_perc, color="Events")) +
  geom_line(stat="identity", aes(y=num_mp_events_perc, color="Events", group=1)) +
  xlab("Event Frequency Max") +
  ylab("%") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
ggsave(filename="mp_events_freq_sum.png")
```

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

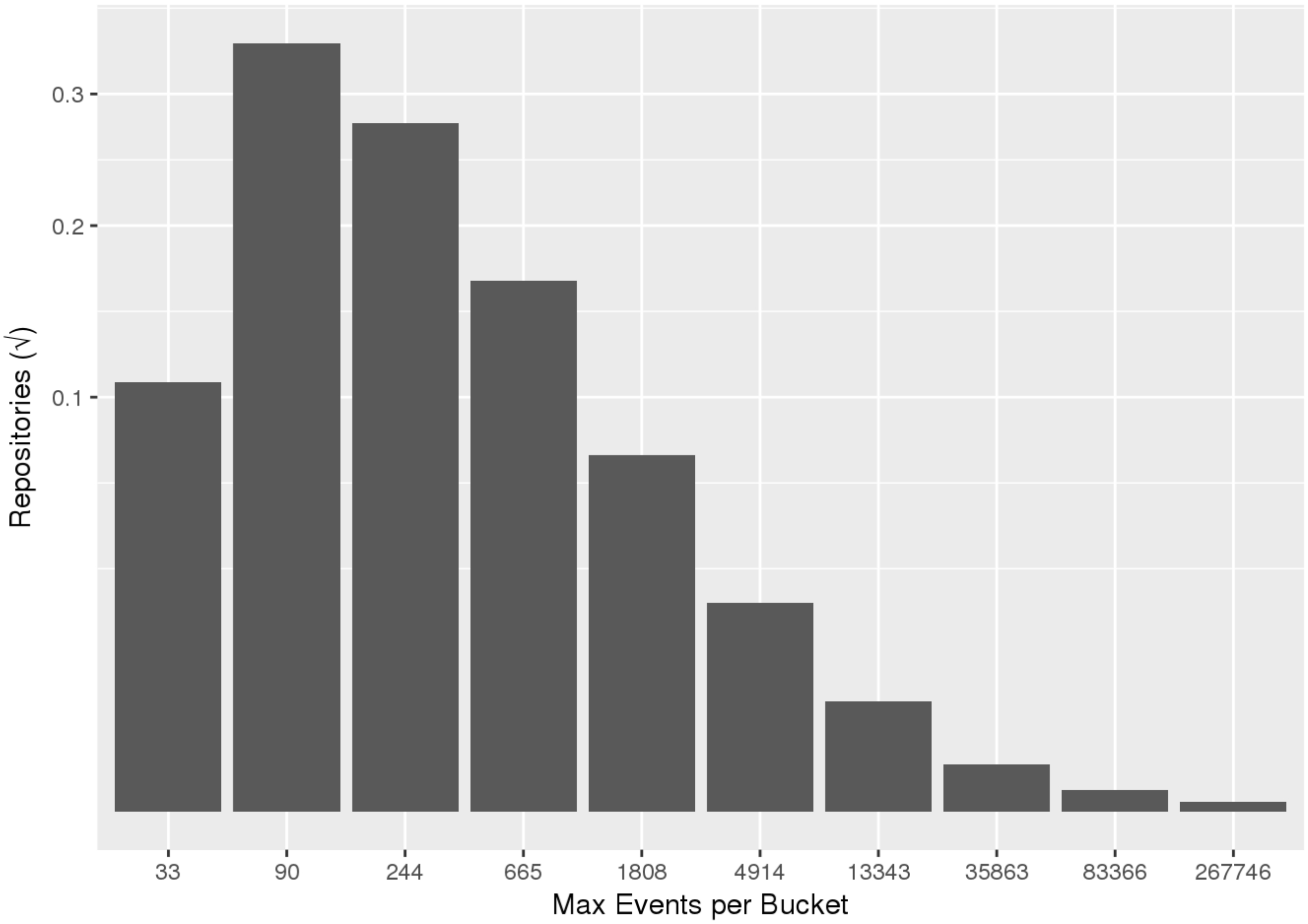
## High Participation

For High Participation repositories, 80% of the repositories had between 34 and 665 events in the 6 month period.

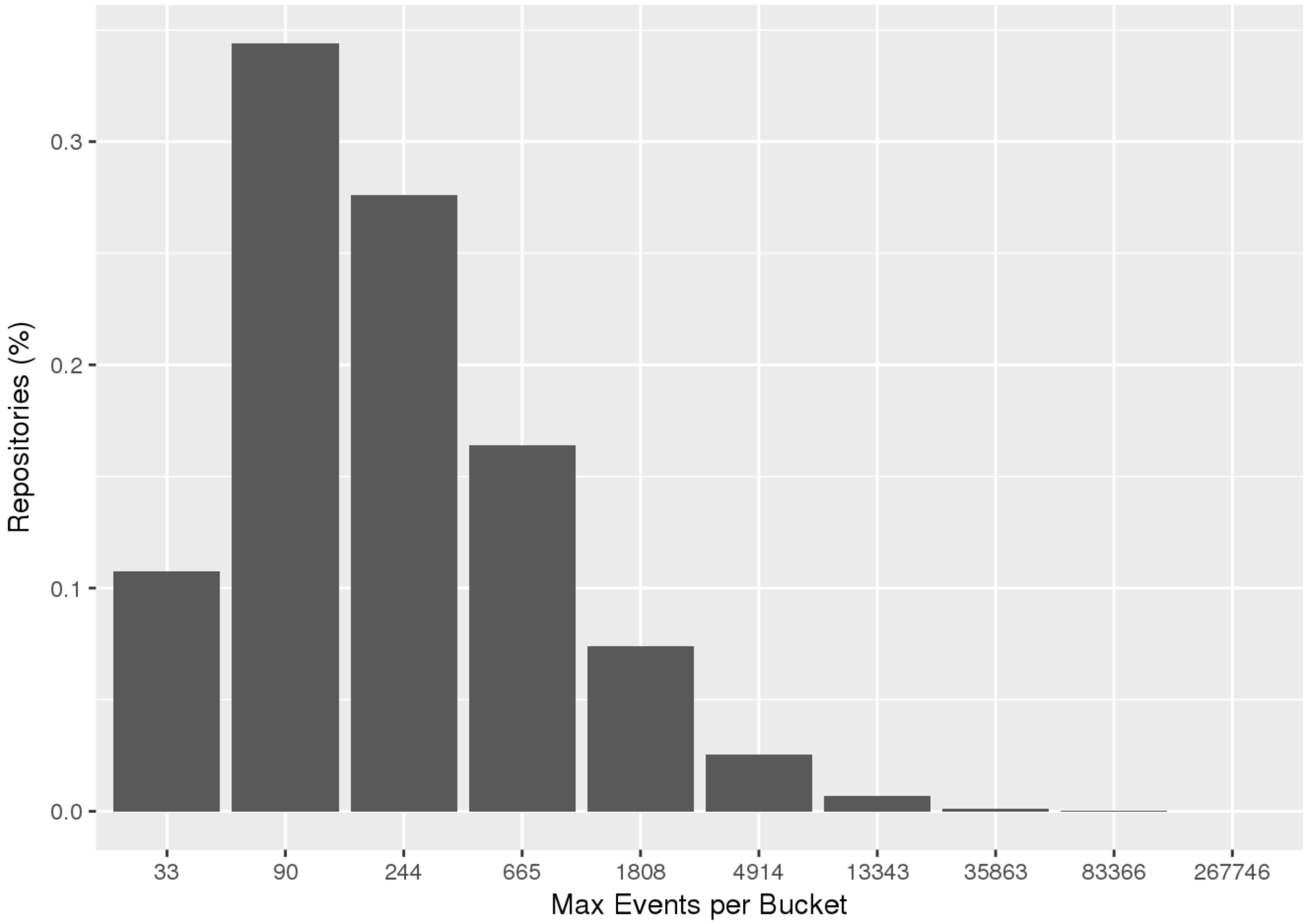
```
hp_summary <- participation_rate_summary %>% filter(participation_level == "High")
```

```
high_participation_num_events_freq <-
readRDS("high_participation_num_events_freq.rds")
high_participation_num_events_freq <- high_participation_num_events_freq %>%
  mutate(num_events_perc = (event_freq*num_repos)/total_events,
         num_hp_events_perc = (event_freq*num_repos)/hp_summary$num_events)
```

```
hp_events_freq_sum <- high_participation_num_events_freq %>%
  group_by(event_freq_log) %>%
  summarise(num_repos = sum(num_repos),
            event_freq_min = min(event_freq),
            event_freq_max = max(event_freq),
            event_freq_med = median(event_freq),
            event_freq_cnt = n(),
            num_hp_repos_perc = num_repos/hp_summary$num_repos,
            num_events_perc = sum(num_events_perc),
            num_hp_events_perc = sum(num_hp_events_perc))

ggplot(data = hp_events_freq_sum, aes(x=factor(event_freq_max), y=num_hp_repos_perc))
+
   geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Repositories (√)") +
  scale_y_continuous(trans = "sqrt")
```

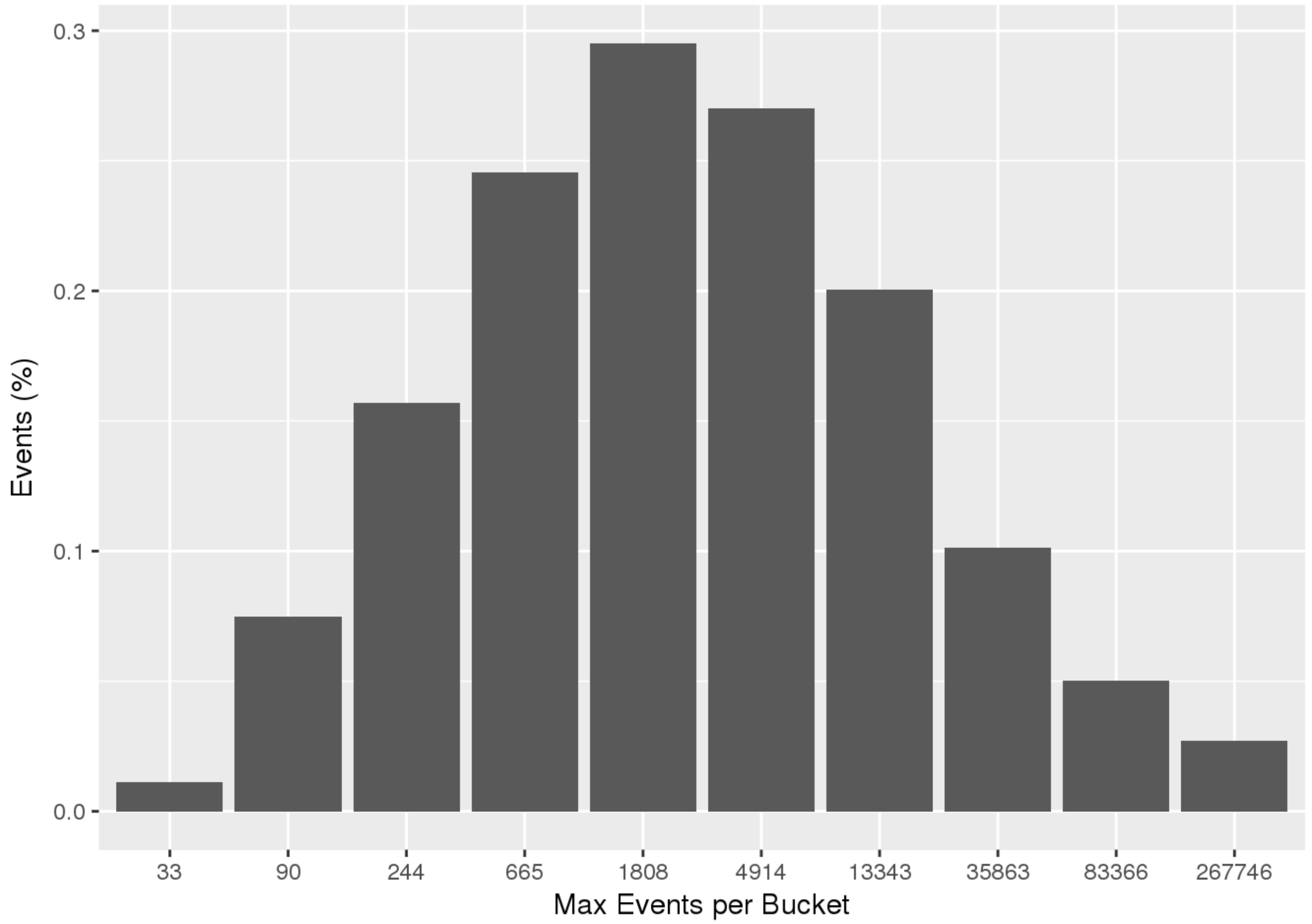


```
ggplot(data = hp_events_freq_sum,
       aes(x=factor(event_freq_max), y=num_hp_repos_perc)) +
  geom_bar(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Repositories (%)")
```

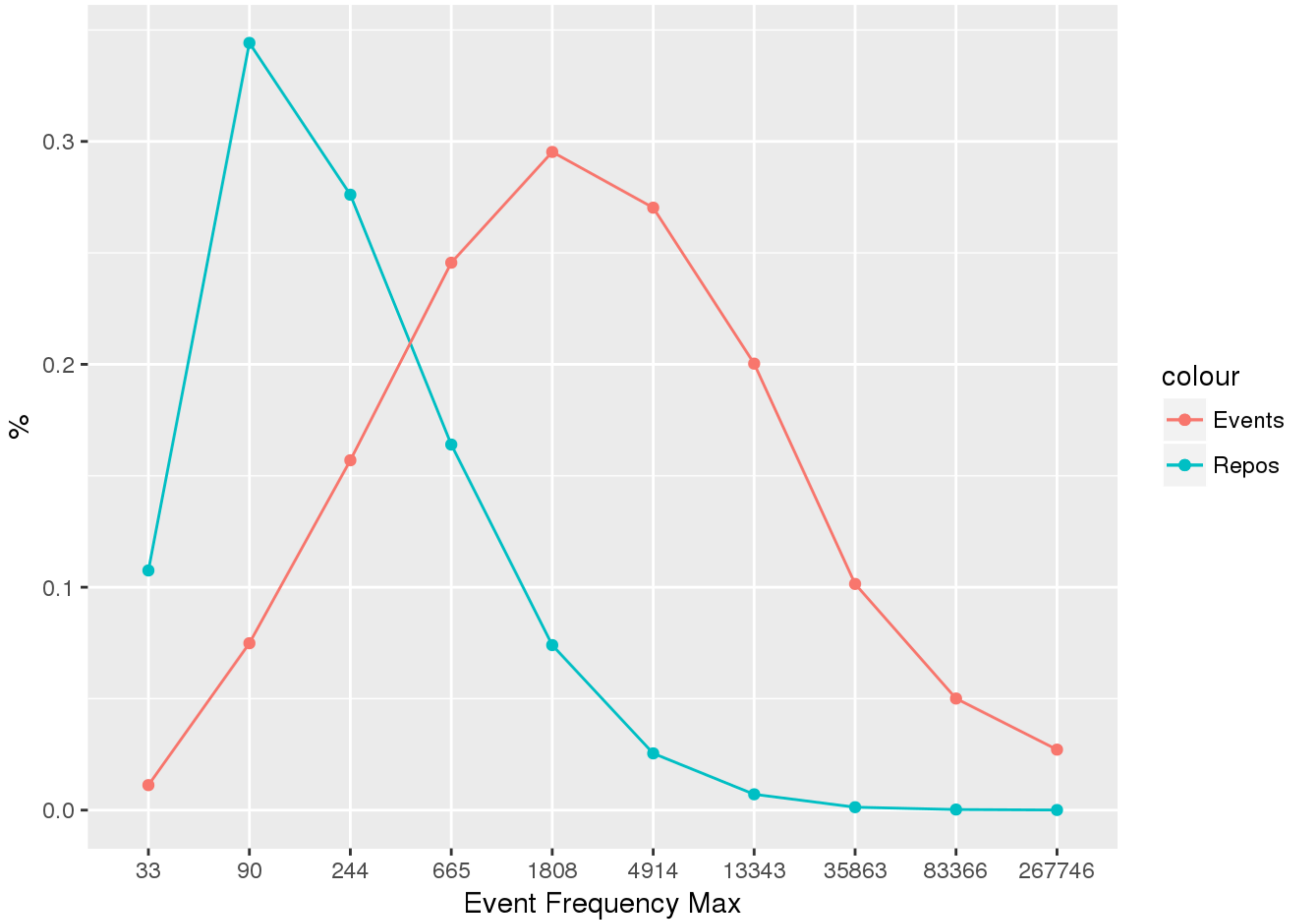


```
ggplot(data = hp_events_freq_sum,
       aes(x=factor(event_freq_max), y=num_hp_events_perc)) +
  geom_histogram(stat="identity") +
  xlab("Max Events per Bucket") +
  ylab("Events (%)")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



```
ggplot(data = hp_events_freq_sum,
       aes(x=factor(event_freq_max))) +
  geom_point(stat="identity", aes(y=num_hp_repos_perc, color="Repos")) +
  geom_line(stat="identity", aes(y=num_hp_repos_perc, color="Repos", group=1)) +
  geom_point(stat="identity", aes(y=num_hp_events_perc, color="Events")) +
  geom_line(stat="identity", aes(y=num_hp_events_perc, color="Events", group=1)) +
  xlab("Event Frequency Max") +
  ylab("%")
```



```
ggsave(filename="hp_events_freq_sum.png")
```

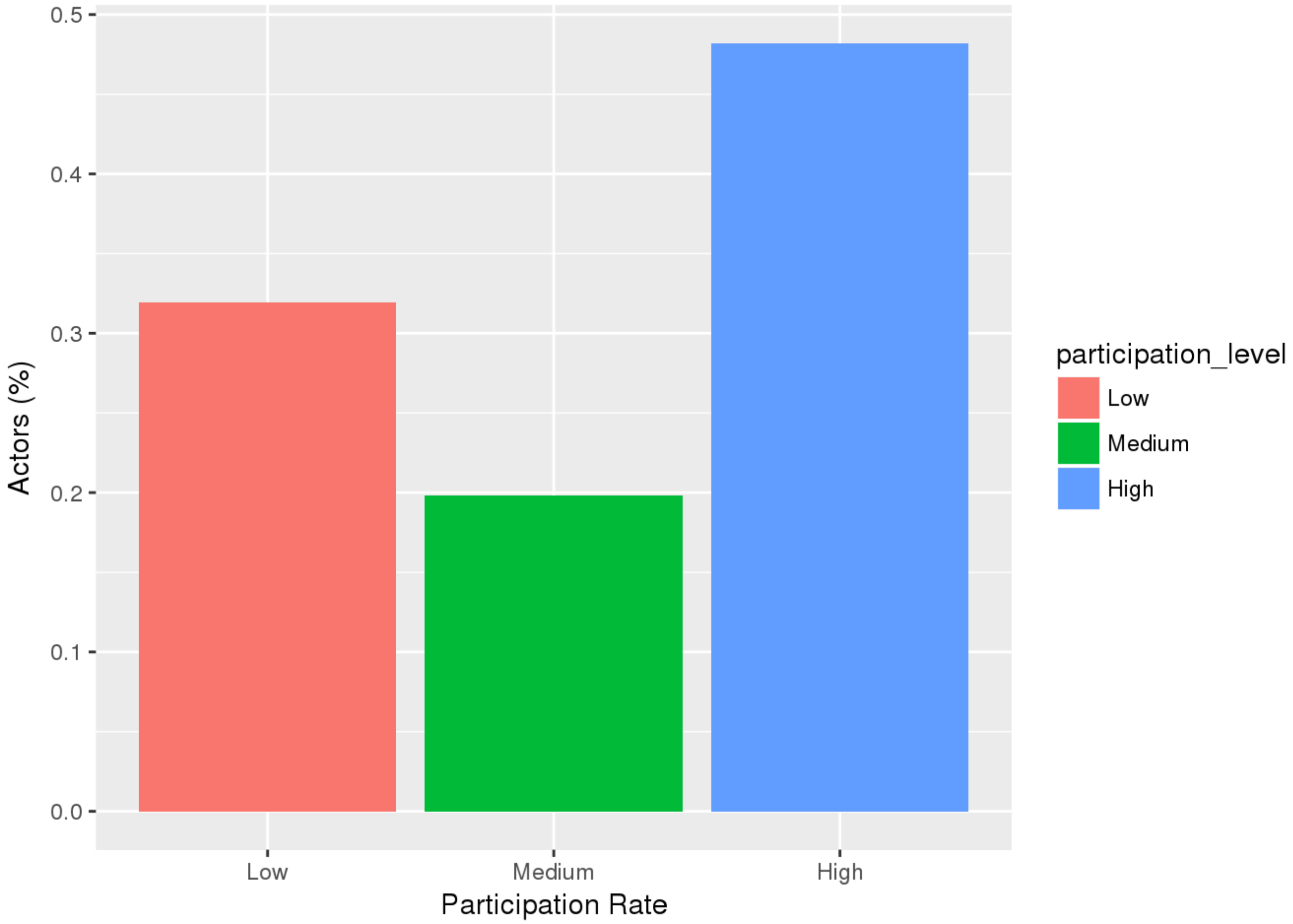
```
## Saving 7 x 5 in image
```

# What was the distribution of actors in each participation level?

## All Levels

Low and High participation repositories had the most unique actors.

```
total_actors <- sum(participation_rate_summary$num_actors)

ggplot(data = participation_rate_summary,
       aes(x=participation_level, y=num_actors/total_actors,
fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Actors (%)")
```

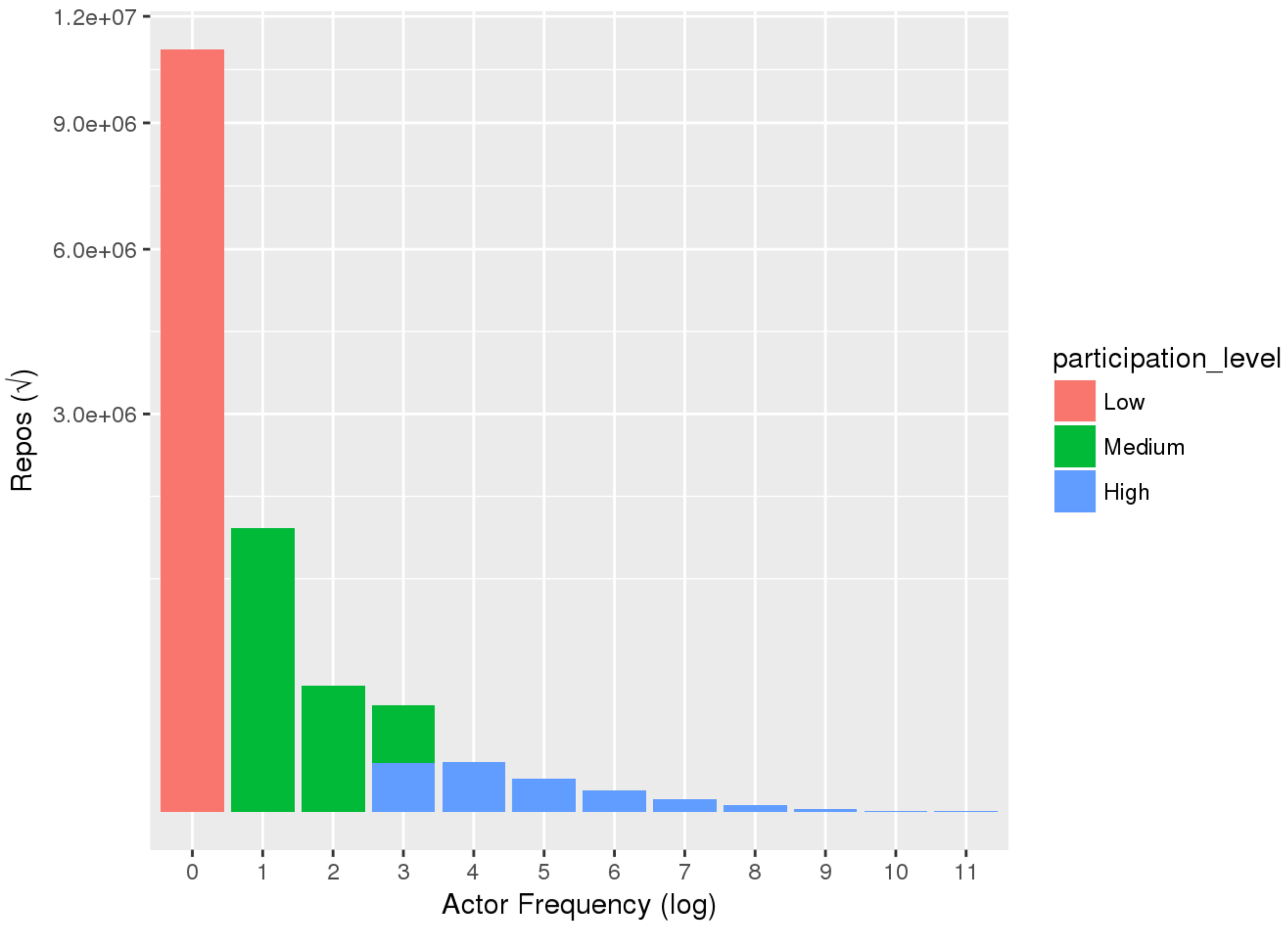


```
ggsave(filename="participation_actors_pct.png")
```

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

```
participation_num_actors_freq <- readRDS("participation_num_actors_freq.rds")

participation_num_actors_freq_sum <- participation_num_actors_freq %>%
  group_by(participation_level, actor_freq_log) %>%
  summarise(num_repos = sum(num_repos),
            actor_freq_min = min(actor_freq),
            actor_freq_max = max(actor_freq),
            actor_freq_med = median(actor_freq),
            actor_freq_cnt = n())
```

```
participation_num_actors_freq_sum$participation_level <- factor(
  participation_num_actors_freq_sum$participation_level,
  levels = unique(participation_num_actors_freq_sum$participation_level[
    order(participation_num_actors_freq_sum$num_repos, decreasing=TRUE)]))

ggplot(data = participation_num_actors_freq_sum,
       aes(x=factor(actor_freq_log), y=num_repos,
           fill = participation_level)) +
  geom_bar(stat="identity") +
  xlab("Actor Frequency (log)") +
  ylab("Repos (√)") +
  scale_y_continuous(trans = "sqrt")
```



```
ggsave(filename="participation_num_actors_freq.png")
```

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

## Low Participation

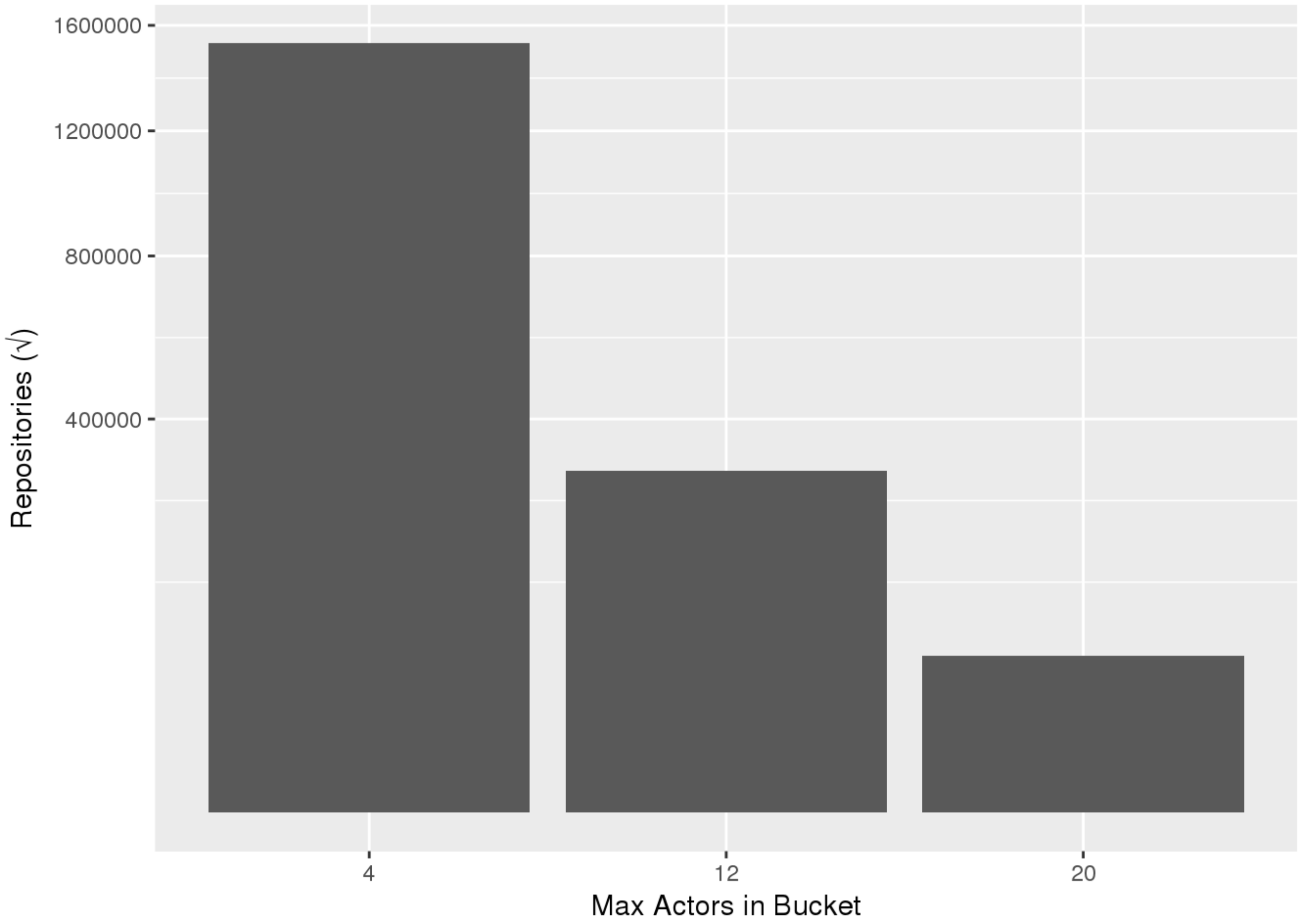
All of the Low participation repositories had only 1 unique actor per repository.

## Medium Participation

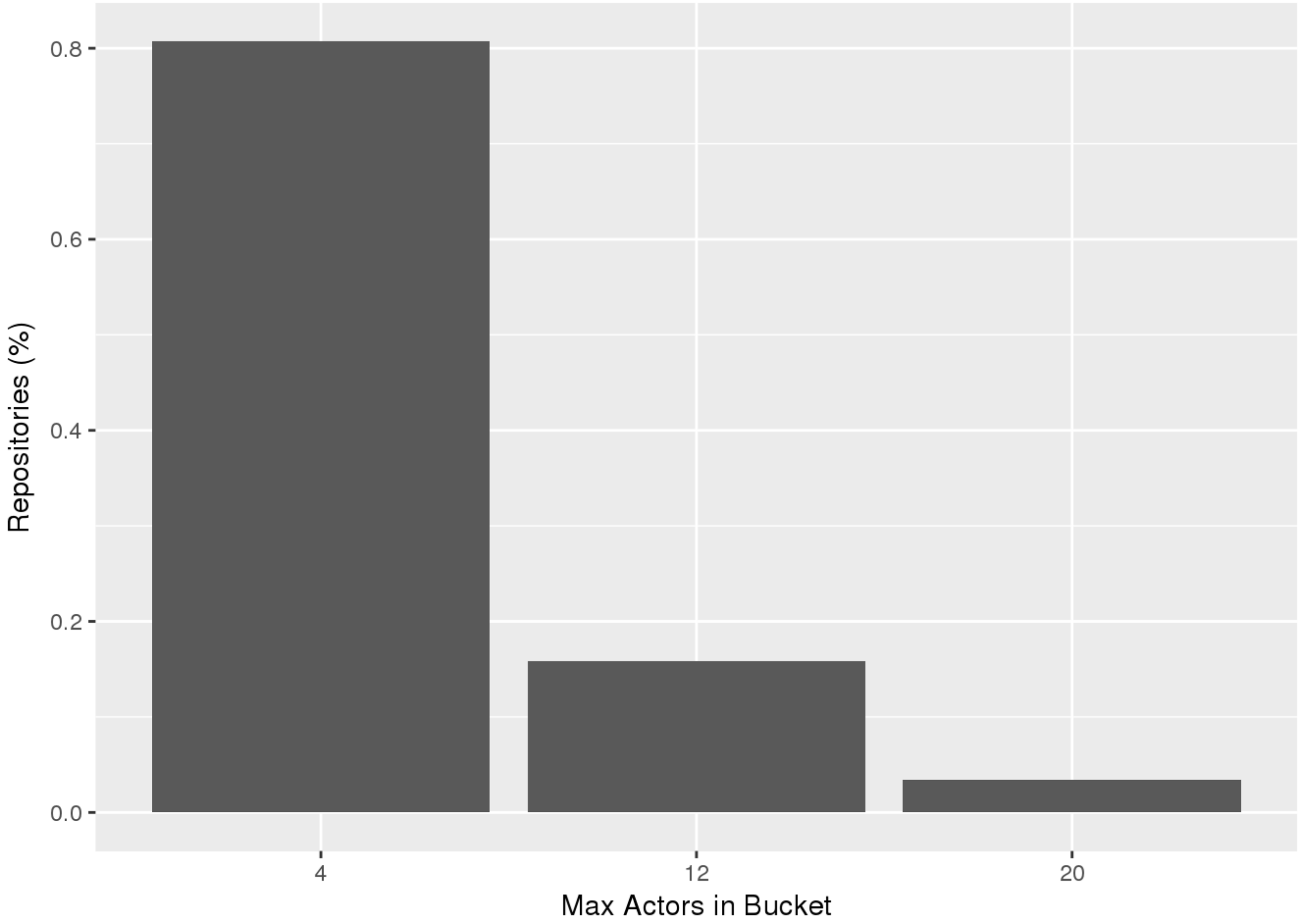
Medium participation repositories had between 2 and 29 unique actors per repository. The majority of these repositories had between 2 to 4 unique actors. 55% of Medium participation repositories had only 2 actors. It’s possible this indicates a problem with the participation rate range used to select this group (.1-.5). Further analysis should compare a split of this range with the High and Low repositories to see if this level is actually necessary or if we can just use a “high” and “low” rating.

```
med_participation_num_actors_freq <- readRDS("med_participation_num_actors_freq.rds")
med_participation_num_actors_freq <- med_participation_num_actors_freq %>%
  mutate(num_actors_perc = (actor_freq*num_repos)/total_actors,
         num_mp_actors_perc = (actor_freq*num_repos)/mp_summary$num_actors,
         num_mp_repos_perc = num_repos/mp_summary$num_repos)
```

```
mp_actors_freq_sum <- med_participation_num_actors_freq %>%
  group_by(actor_freq_log) %>%
  summarise(num_repos = sum(num_repos),
            actor_freq_min = min(actor_freq),
            actor_freq_max = max(actor_freq),
            actor_freq_med = median(actor_freq),
            actor_freq_cnt = n(),
            num_mp_repos_perc = sum(num_mp_repos_perc),
            num_actors_perc = sum(num_actors_perc),
            num_mp_actors_perc = sum(num_mp_actors_perc))

ggplot(data = mp_actors_freq_sum, aes(x=factor(actor_freq_max), y=num_repos)) +
   geom_bar(stat="identity") +
  xlab("Max Actors in Bucket") +
  ylab("Repositories (√)") +
  scale_y_continuous(trans = "sqrt")
```



```
ggplot(data = mp_actors_freq_sum,
       aes(x=factor(actor_freq_max), y=num_mp_repos_perc)) +
  geom_bar(stat="identity") +
  xlab("Max Actors in Bucket") +
  ylab("Repositories (%)")
```



```
ggsave(filename="mp_actors_freq_sum.png")
```

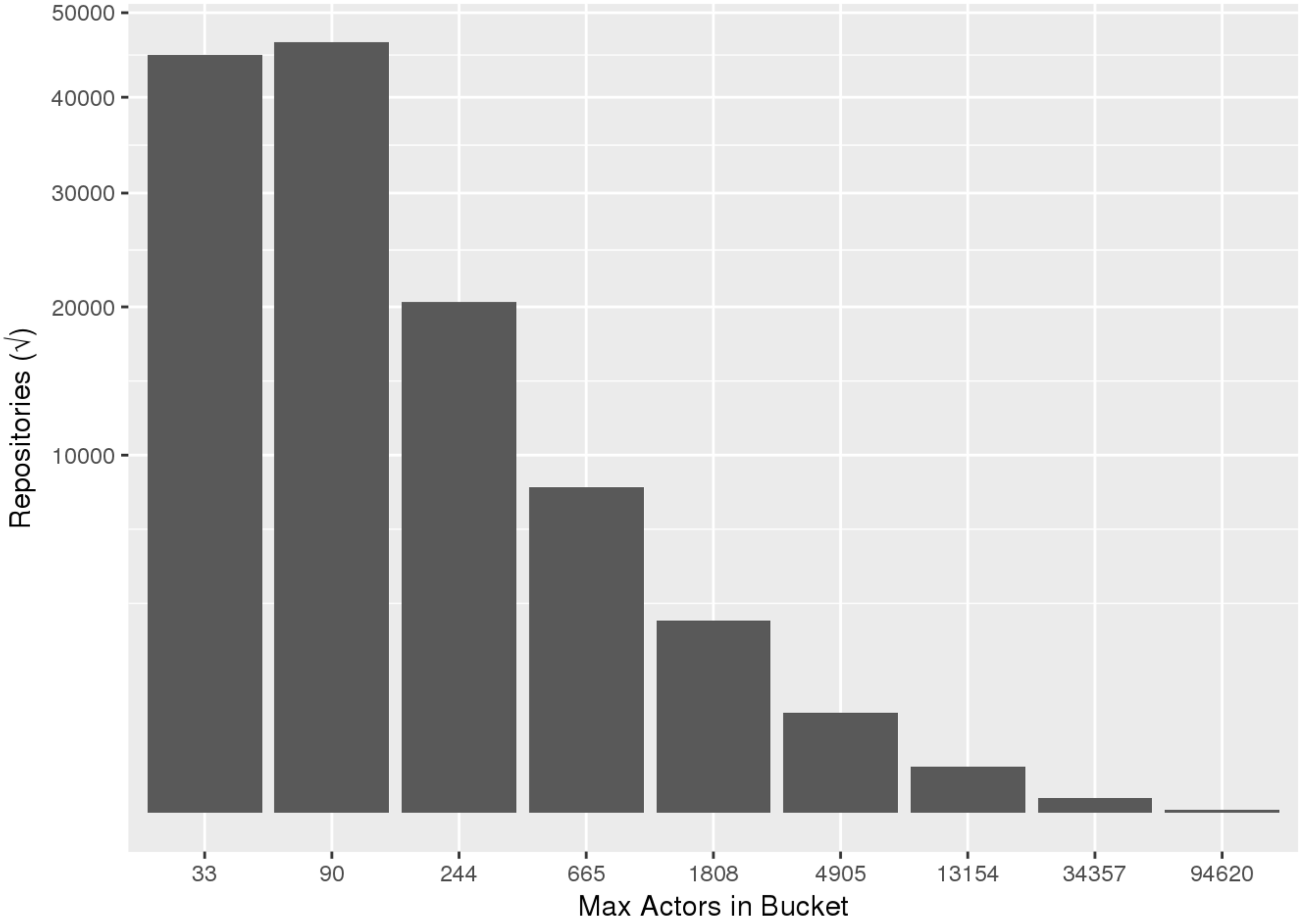
```
## Saving 7 x 5 in image
```

## High Participation

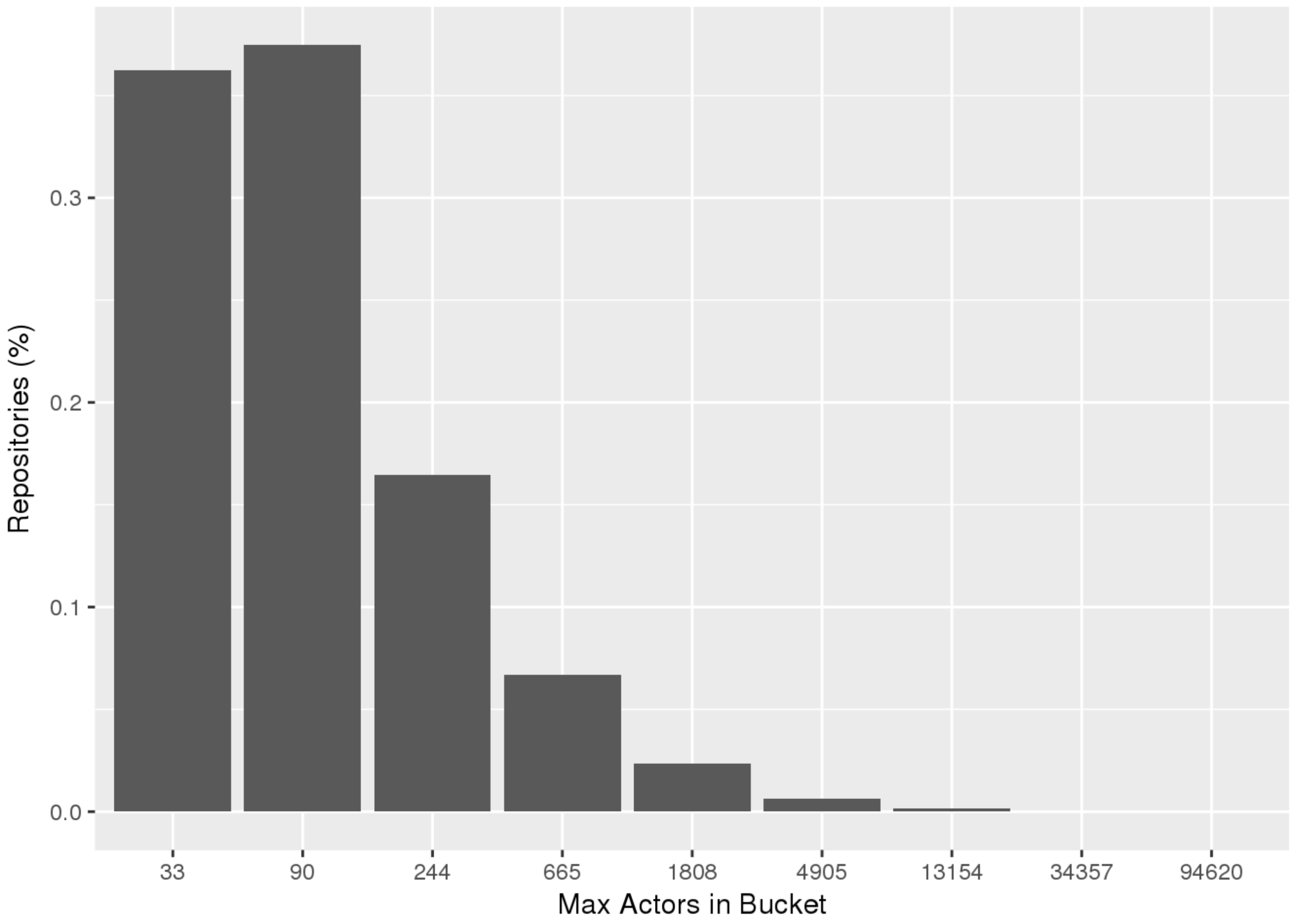
The majority of High Participation repositories (~70%) have between 30 and 90 unique actors. The rest have a really large spread going over 90,000 for one repository. This distribution further supports what was shown previously, that the participation rates assigned to each level might need to be shifted. Further analysis looking at samples will give us more insight into how to do this.

```
high_participation_num_actors_freq <-
readRDS("high_participation_num_actors_freq.rds")
high_participation_num_actors_freq <- high_participation_num_actors_freq %>%
  mutate(num_actors_perc = (actor_freq*num_repos)/total_actors,
         num_hp_actors_perc = (actor_freq*num_repos)/hp_summary$num_actors,
         num_hp_repos_perc = num_repos/hp_summary$num_repos)
```

```
hp_actors_freq_sum <- high_participation_num_actors_freq %>%
  group_by(actor_freq_log) %>%
  summarise(num_repos = sum(num_repos),
            actor_freq_min = min(actor_freq),
            actor_freq_max = max(actor_freq),
            actor_freq_med = median(actor_freq),
            actor_freq_cnt = n(),
            num_hp_repos_perc = sum(num_hp_repos_perc),
            num_actors_perc = sum(num_actors_perc),
            num_hp_actors_perc = sum(num_hp_actors_perc))

ggplot(data = hp_actors_freq_sum, aes(x=factor(actor_freq_max), y=num_repos)) +
   geom_bar(stat="identity") +
  xlab("Max Actors in Bucket") +
  ylab("Repositories (√)") +
  scale_y_continuous(trans = "sqrt")
```



```
ggplot(data = hp_actors_freq_sum,
       aes(x=factor(actor_freq_max), y=num_hp_repos_perc)) +
  geom_bar(stat="identity") +
  xlab("Max Actors in Bucket") +
  ylab("Repositories (%)")
```



```
ggsave(filename="hp_actors_freq_sum.png")
```

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

# How does the frequency of event types compare between participation rates?

```
participation_rate_event_types <- readRDS("participation_rate_event_types.rds")
participation_rate_event_types <- participation_rate_event_types %>%
  mutate(events_log = round(log(num_events)),
         actors_log = round(log(num_actors)))
```

## Events

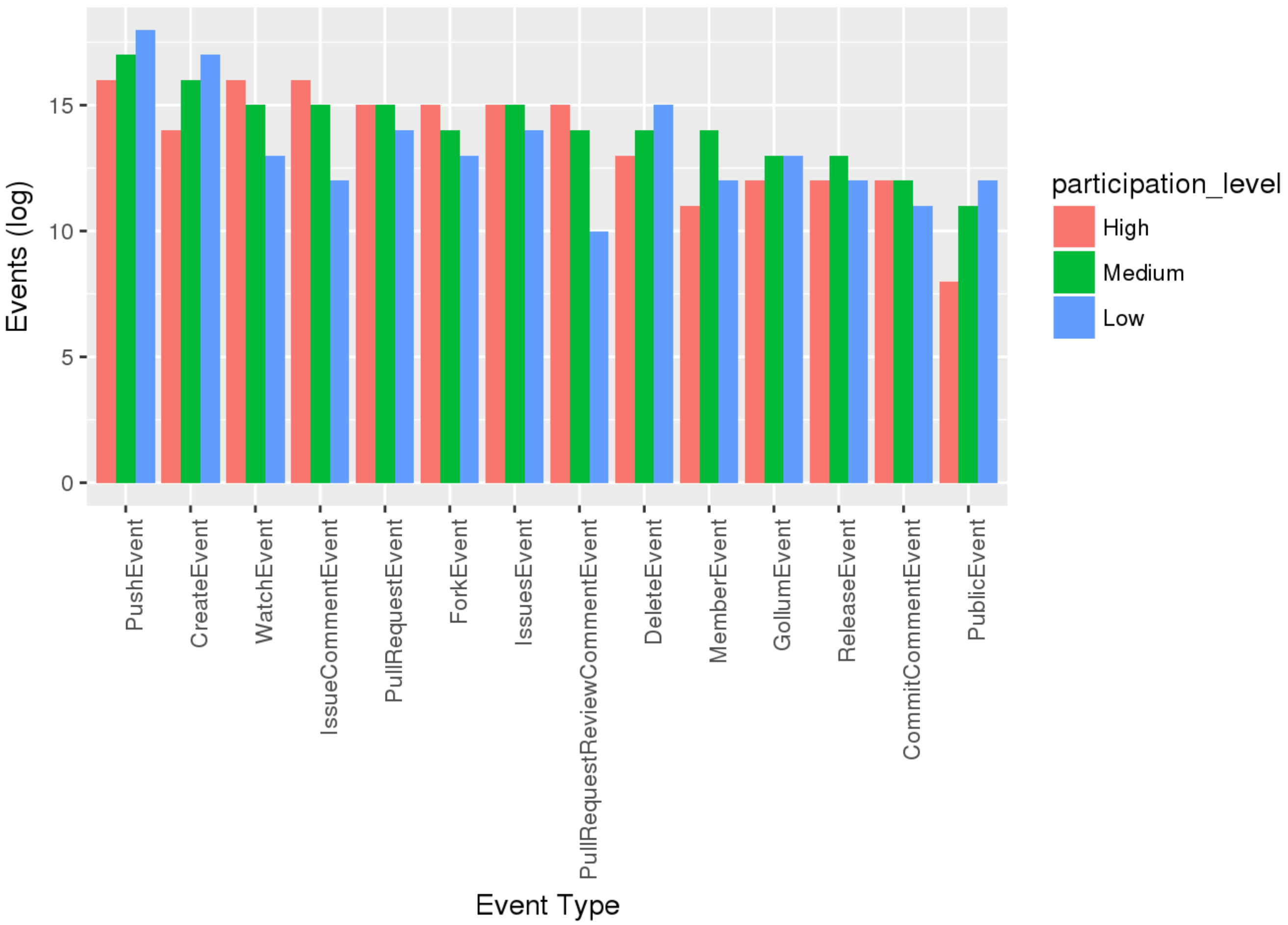
Some event types are fairly evenly distributed between the participation rates while others are not.

This preliminary analysis allows us to propose the following:

1. Samples taken from Push events would show the most variation and diversity.
2. Create events show the highest association with Medium and Low repositories.
3. Watch, Issue Comment, and Fork events occur most frequently with High participation repositories.

Finally, Medium repositories do not appear to have distinct characteristics from Low or High repositories. Future analysis should compare splitting Medium between High and Low repositories and recalculating the participation rate and readjusting the ranges represented by each group.

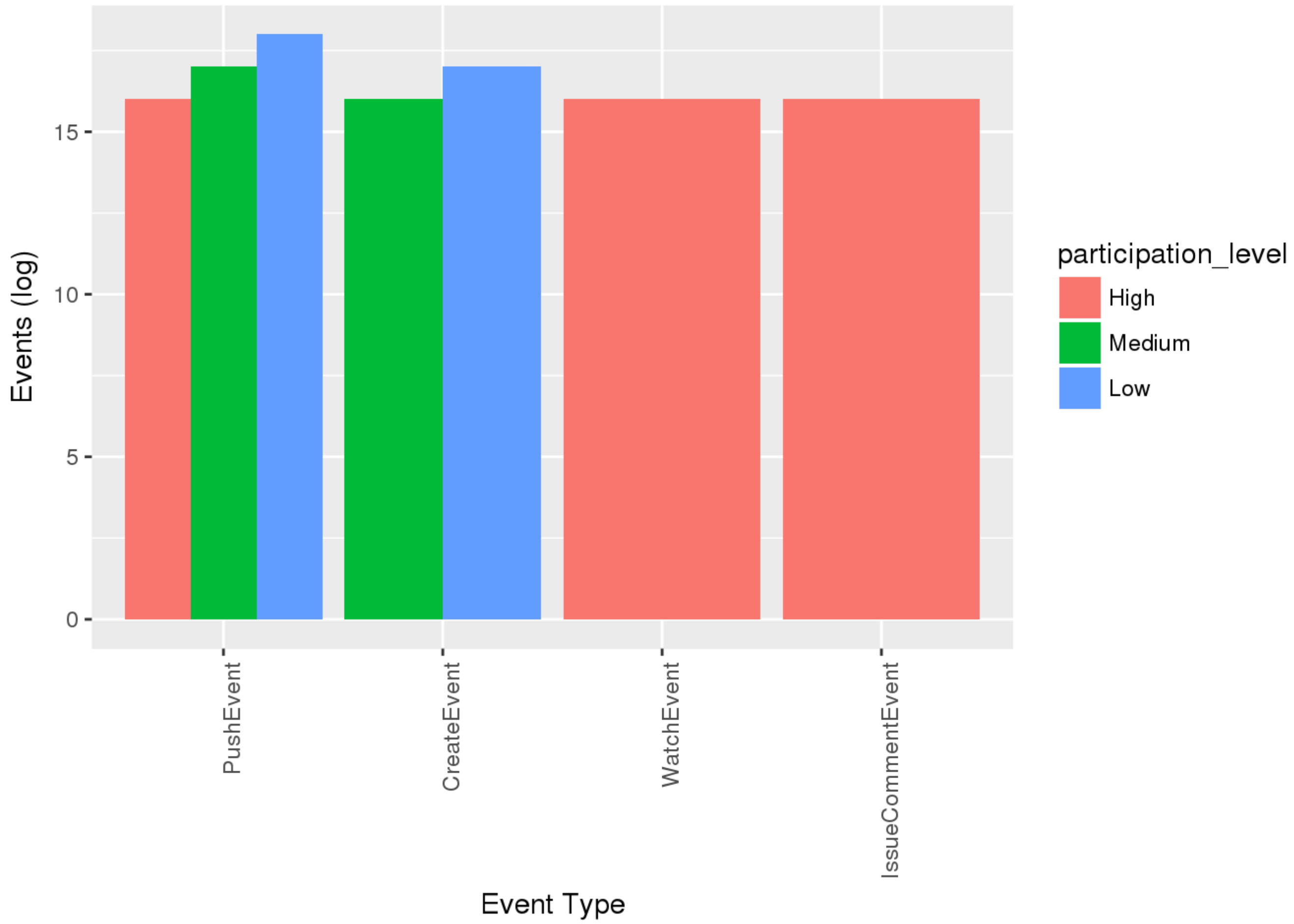
```
participation_rate_event_types$type <-
  factor(participation_rate_event_types$type,
         levels = unique(participation_rate_event_types$type[
           order(participation_rate_event_types$num_events, decreasing=TRUE)]))

participation_rate_event_types$participation_level <-
  factor(participation_rate_event_types$participation_level,
  levels = c("High", "Medium", "Low"))

ggplot(data = participation_rate_event_types,
       aes(x=type,
       y=events_log,
       fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  ylab("Events (log)") +
  xlab("Event Type") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
ggsave("participation_event_types.png")
```

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

```
pr_max_events <- participation_rate_event_types %>%
  filter( (participation_level == "High" & events_log >= max(events_log)) | (
            (participation_level == "Medium" | participation_level == "Low") &
              events_log >= max(events_log)-1))

ggplot(data = pr_max_events,
       aes(x=type,
       y=events_log,
       fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  ylab("Events (log)") +
  xlab("Event Type") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



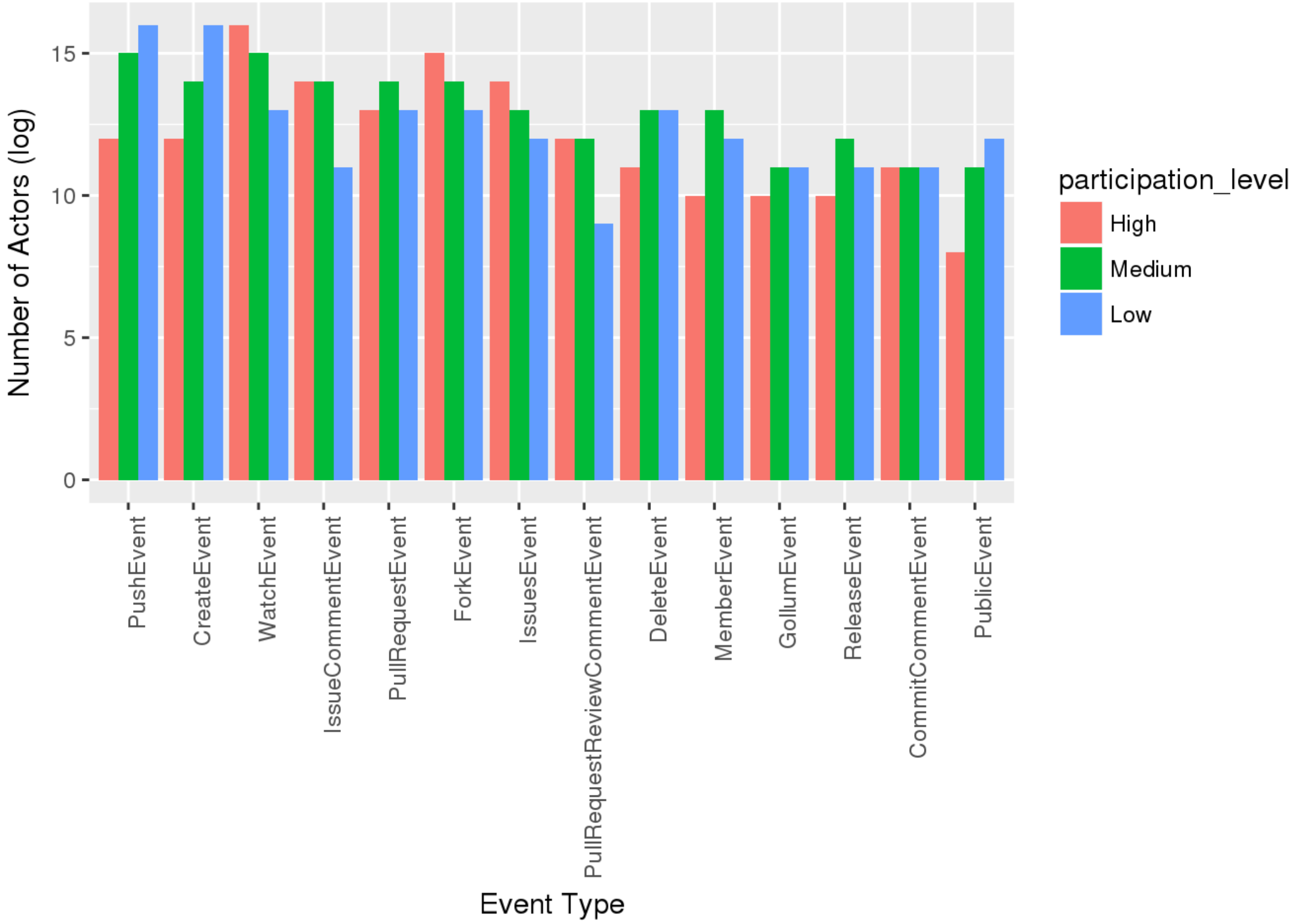
```
ggsave("participation_max_events.png")
```

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

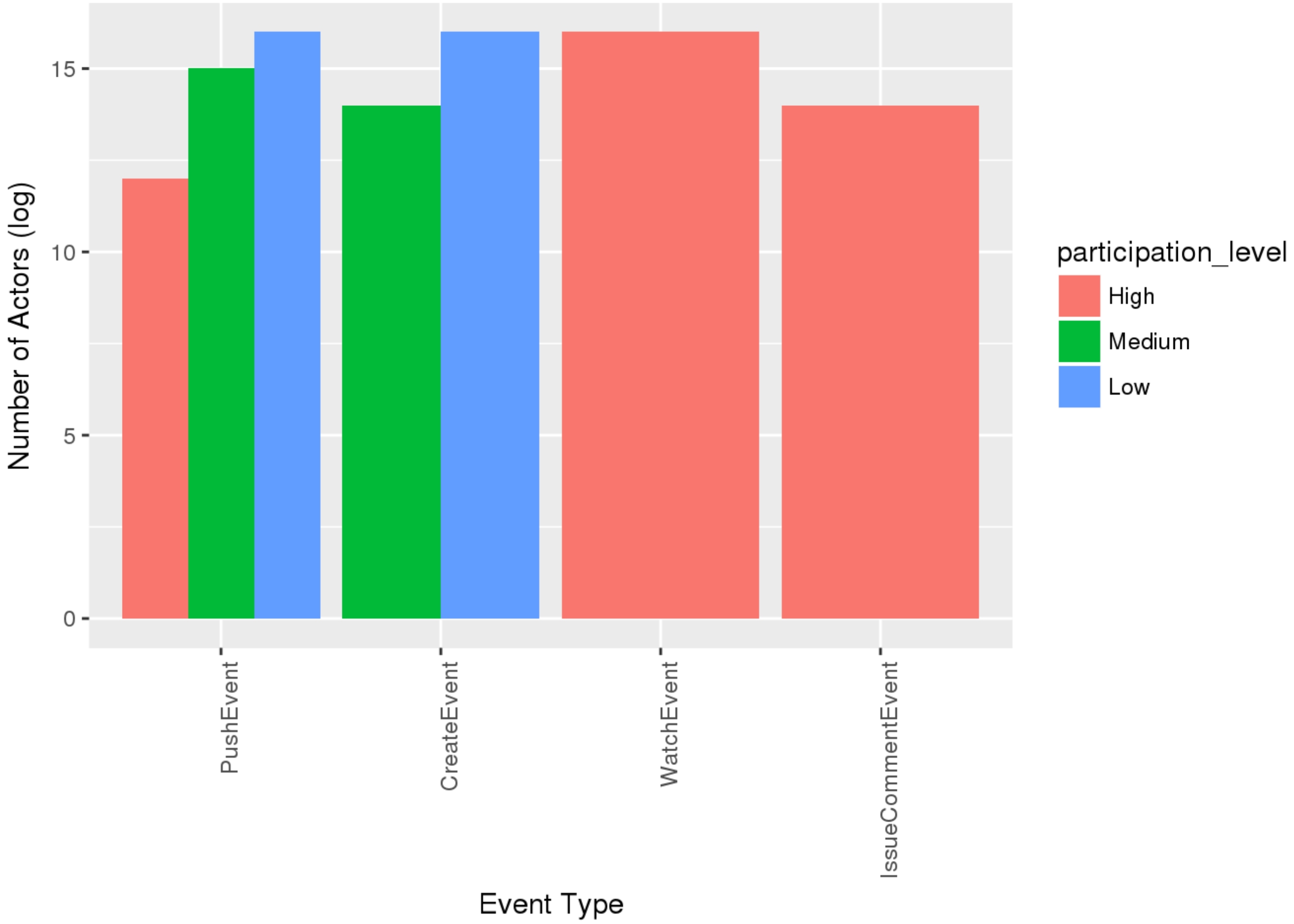
## Actors

The following plots look at the number of actors across the different event types and in the most frequent event types identified in the previous section. It appears that the most frequent events also had the highest number of unique actors, which would be expected. Medium is spread across both High and Low which is also expected. Looking at event types with the highest number of actors, no new event types show up here suggesting a consistent correlation.

```
ggplot(data = participation_rate_event_types,
       aes(x=type,
       y=actors_log,
       fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  ylab("Number of Actors (log)") +
  xlab("Event Type") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



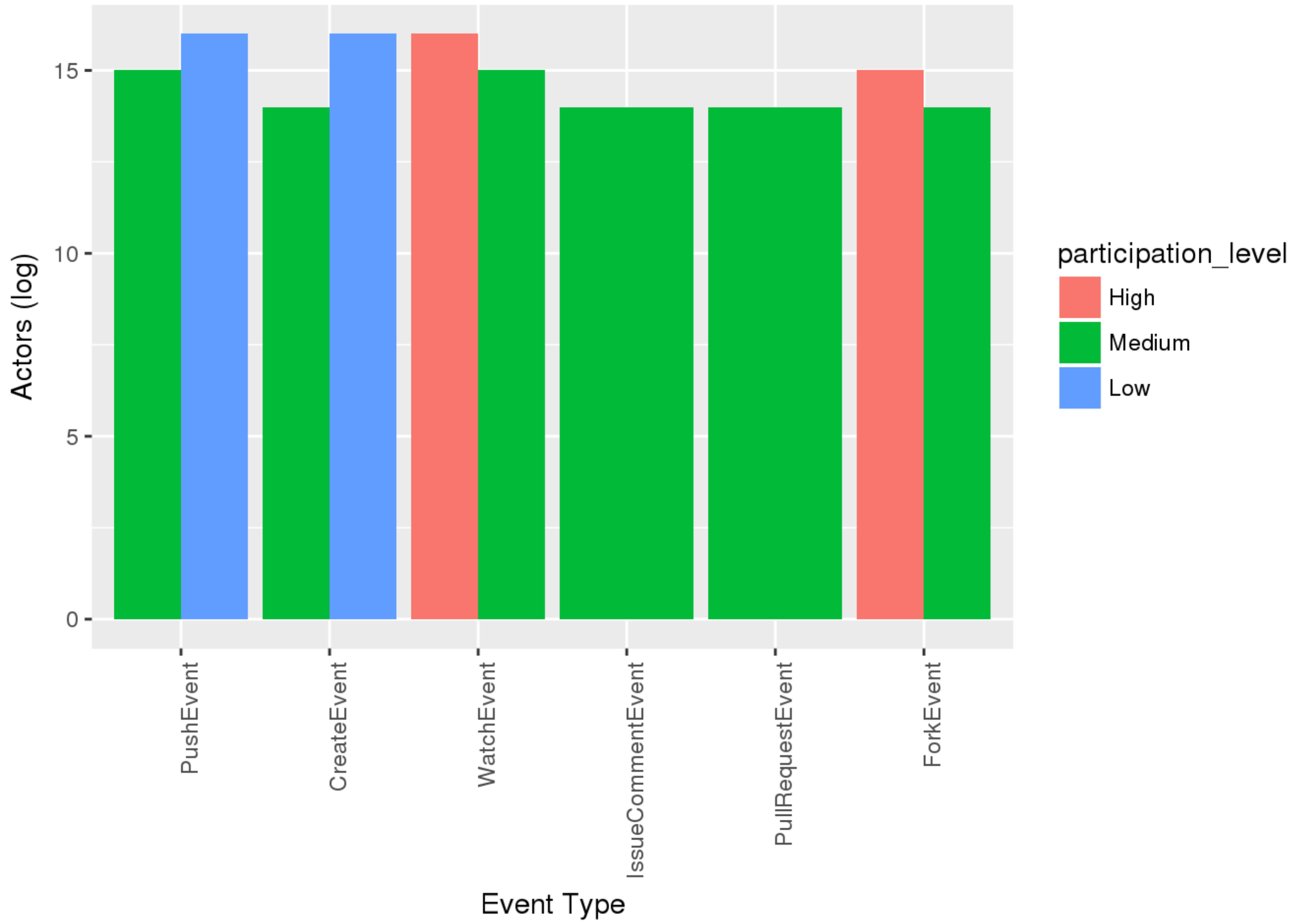
```
ggplot(data = pr_max_events,
       aes(x=type,
       y=actors_log,
       fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  ylab("Number of Actors (log)") +
  xlab("Event Type") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
ggsave(filename="pr_max_events_actors.png")
```

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

```
pr_max_actors <- participation_rate_event_types %>%
  filter(actors_log >= max(actors_log)-1)

ggplot(data = pr_max_actors,
       aes(x=type,
       y=actors_log,
       fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  ylab("Actors (log)") +
  xlab("Event Type") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
ggsave(filename="pr_max_actors.png")
```

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

# Sampling Experiments

In this experiment, samples are pulled from event types that had the highest association with one participation level. The participation level distribution will be evaluated to see if by sampling events of a certain type we can guarantee a majority of repositories at the desired participation level. If this turns out to be the case, it means we do not have to calculate the participation levels of repositories and could potentially just pull samples directly from events in the archive.

# All Participation Levels (Push Events)

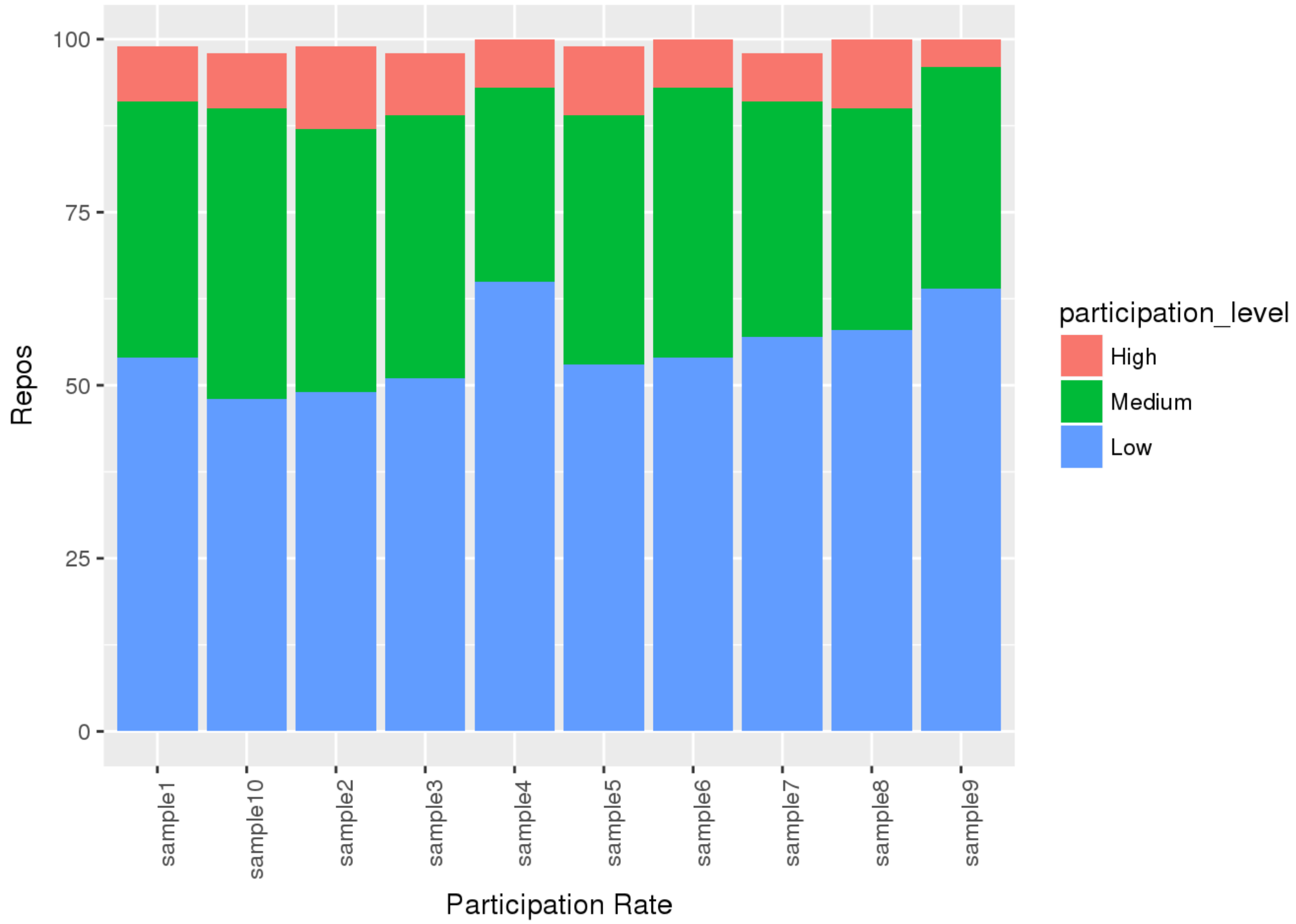
Repositories from Push event samples would provide the best distribution of all participation rates and therefore most closely mimic the population.

```
push_events_repo_samples <- readRDS("push_events_repo_samples.rds")

push_events_repo_samples <-
    mutate(push_events_repo_samples,
           participation_level = ifelse(participation_rate < 1 & participation_rate >
0, 'Medium', ''))

push_events_repo_samples <-
  mutate(push_events_repo_samples, participation_level = ifelse(participation_rate ==
0, 'High', participation_level))

push_events_repo_samples <-
  mutate(push_events_repo_samples, participation_level = ifelse(participation_rate ==
1, 'Low', participation_level))
```

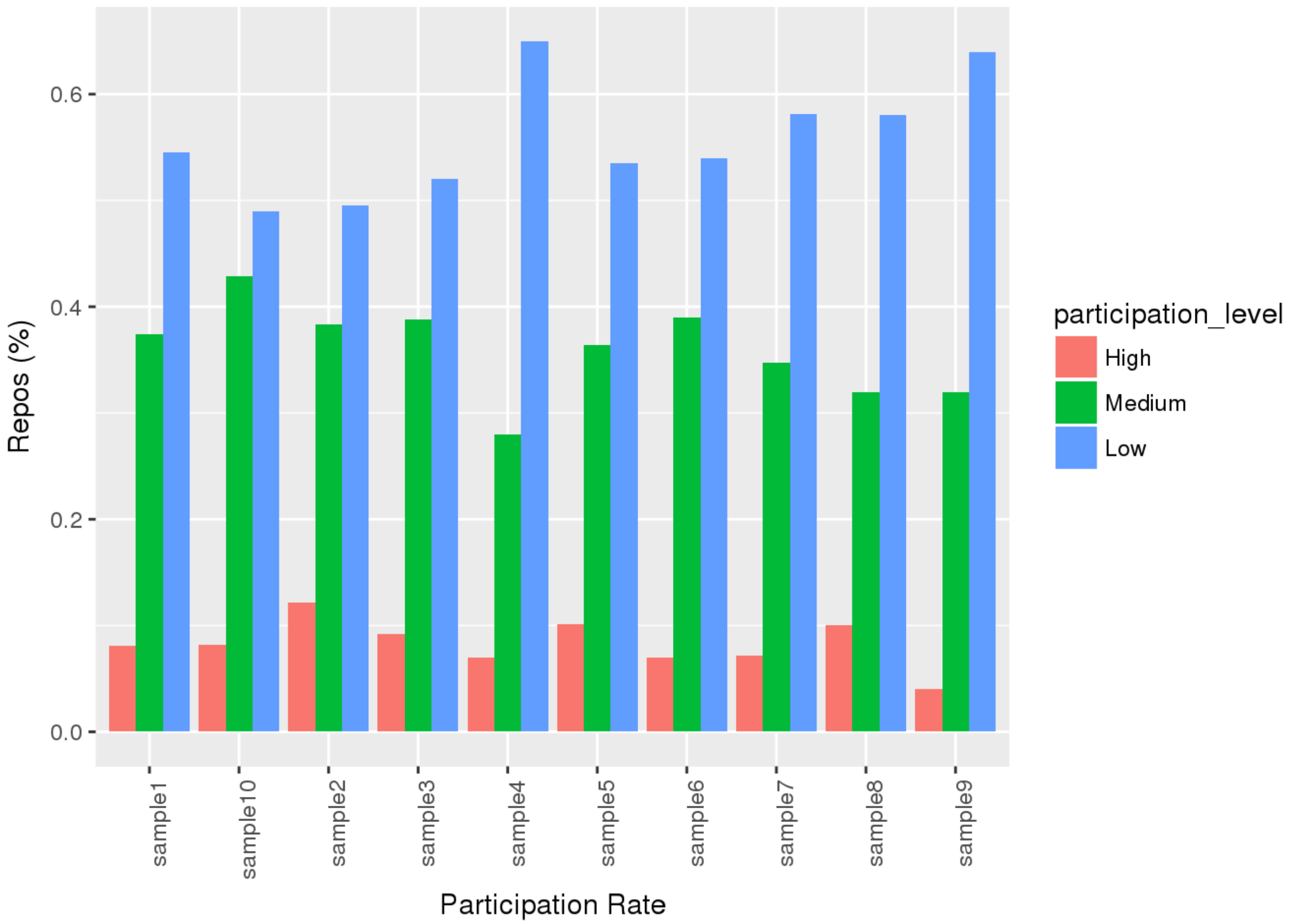
## Participation Rates

```
push_repo_summary <- push_events_repo_samples %>%
  group_by(dataset, repo_name) %>%
  summarise(
    participation_level = max(participation_level),
    num_repo_events = max(num_repo_events),
    num_repo_actors = max(num_repo_actors),
    repo_actors_log = round(log(num_repo_actors)),
    repo_events_log = round(log(num_repo_events))
  )

push_dataset_summary <- push_repo_summary %>%
  group_by(dataset) %>%
  summarise(repos_in_dataset = n(),
            actors_in_dataset = sum(num_repo_actors),
            events_in_dataset = sum(num_repo_events))

push_participation_summary <- push_repo_summary %>%
  group_by(dataset, participation_level) %>%
  summarise(num_repos = n())

push_participation_summary <- merge(push_participation_summary, push_dataset_summary,
by="dataset")

push_participation_summary <- push_participation_summary %>%
  mutate(repos_perc = num_repos/repos_in_dataset)

push_participation_summary$participation_level <- factor(push_participation_summary$pa
rticipation_level,
                                                         levels = c("High", "Medium",
"Low"))
```

```
ggplot(data = push_participation_summary,
       aes(x=dataset, y=num_repos, fill=participation_level)) +
  geom_bar(stat="identity", position="stack") +
  xlab("Participation Rate") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



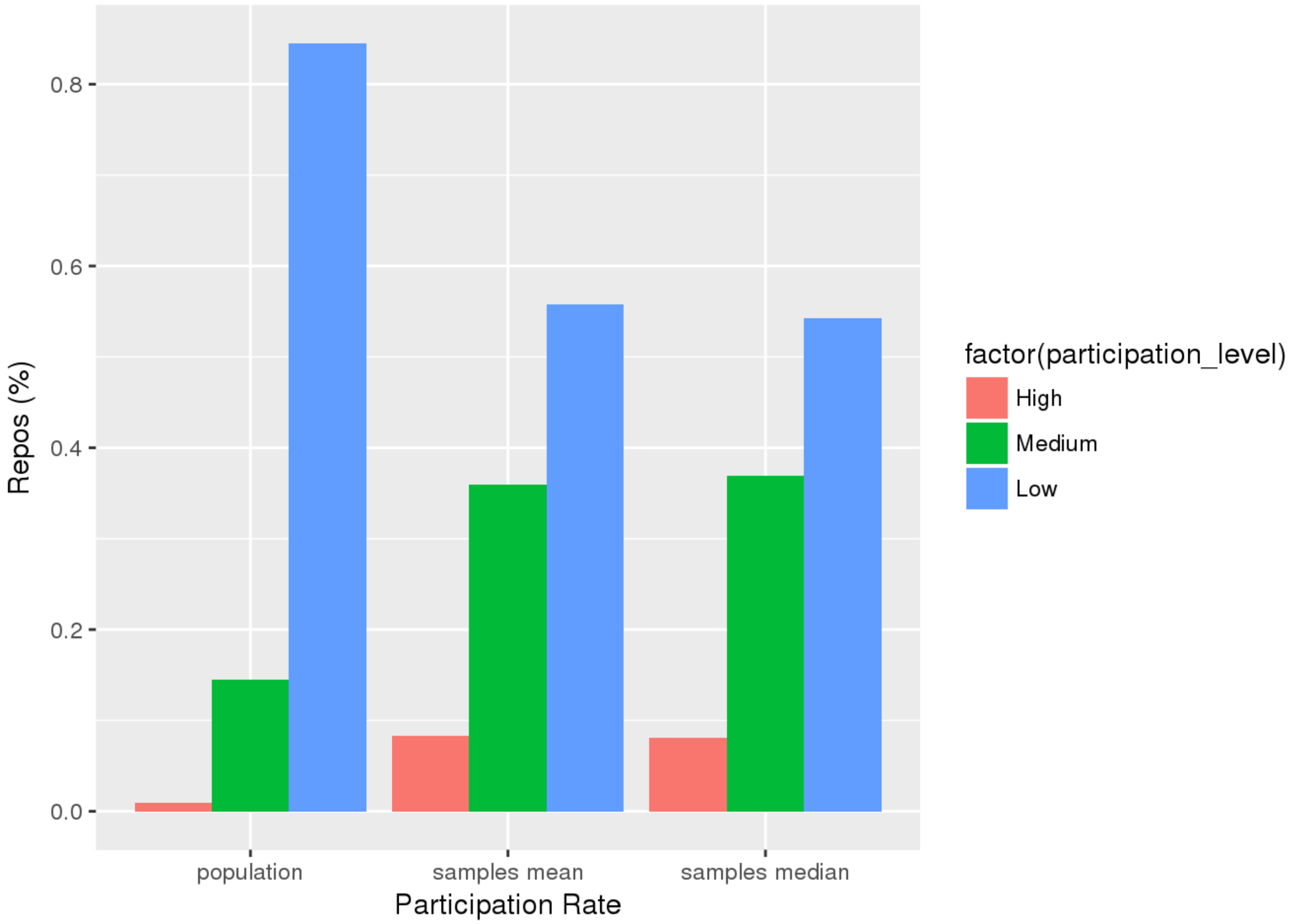
```
ggplot(data = push_participation_summary,
       aes(x=dataset, y=repos_perc, fill=participation_level)) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```


Repos (%)
0.6
0.4
0.2
0.0
sample1
sample10
sample2
sample3
sample4
sample5
sample6
sample7
sample8
sample9
Participation Rate
participation_level
High
Medium
Low

```
participation_rate_summary <- readRDS("participation_rate_summary.rds")
total_repos <- sum(participation_rate_summary$num_repos)
total_events <- sum(participation_rate_summary$num_events)
total_actors <- sum(participation_rate_summary$num_actors)

push_vs_pop_pop <- participation_rate_summary %>%
  mutate(
    dataset = "population",
    repos_perc = num_repos/total_repos
  ) %>%
  select(dataset,
    participation_level,
    repos_perc)

push_vs_pop_push_mean <- push_participation_summary %>%
  group_by(participation_level) %>%
  summarise( dataset = "samples mean",
    repos_perc = mean(repos_perc))

push_vs_pop_push_med <- push_participation_summary %>%
  group_by(participation_level) %>%
  summarise( dataset = "samples median",
    repos_perc = median(repos_perc))

push_vs_pop_participation <- bind_rows(push_vs_pop_push_mean, push_vs_pop_push_med, pu
sh_vs_pop_pop)
```

```
## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector
```

```
push_vs_pop_participation$participation_level <- factor(push_vs_pop_participation$part
icipation_level,
                           levels = c("High", "Medium", "Low"))

ggplot(data = push_vs_pop_participation,
       aes(x=dataset, y=repos_perc, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")
```



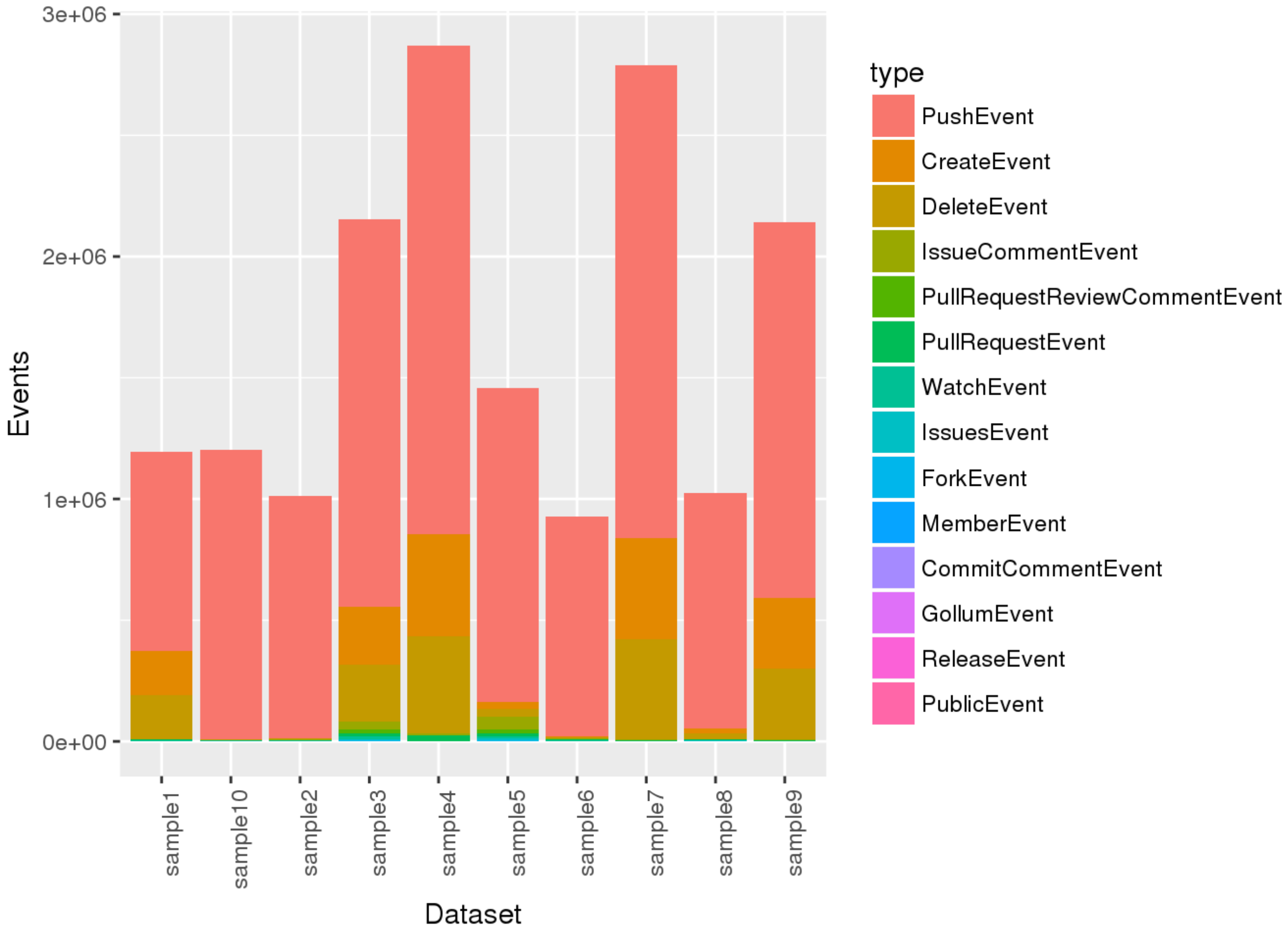
```
ggsave(filename="push_vs_pop_participation.png")
```

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

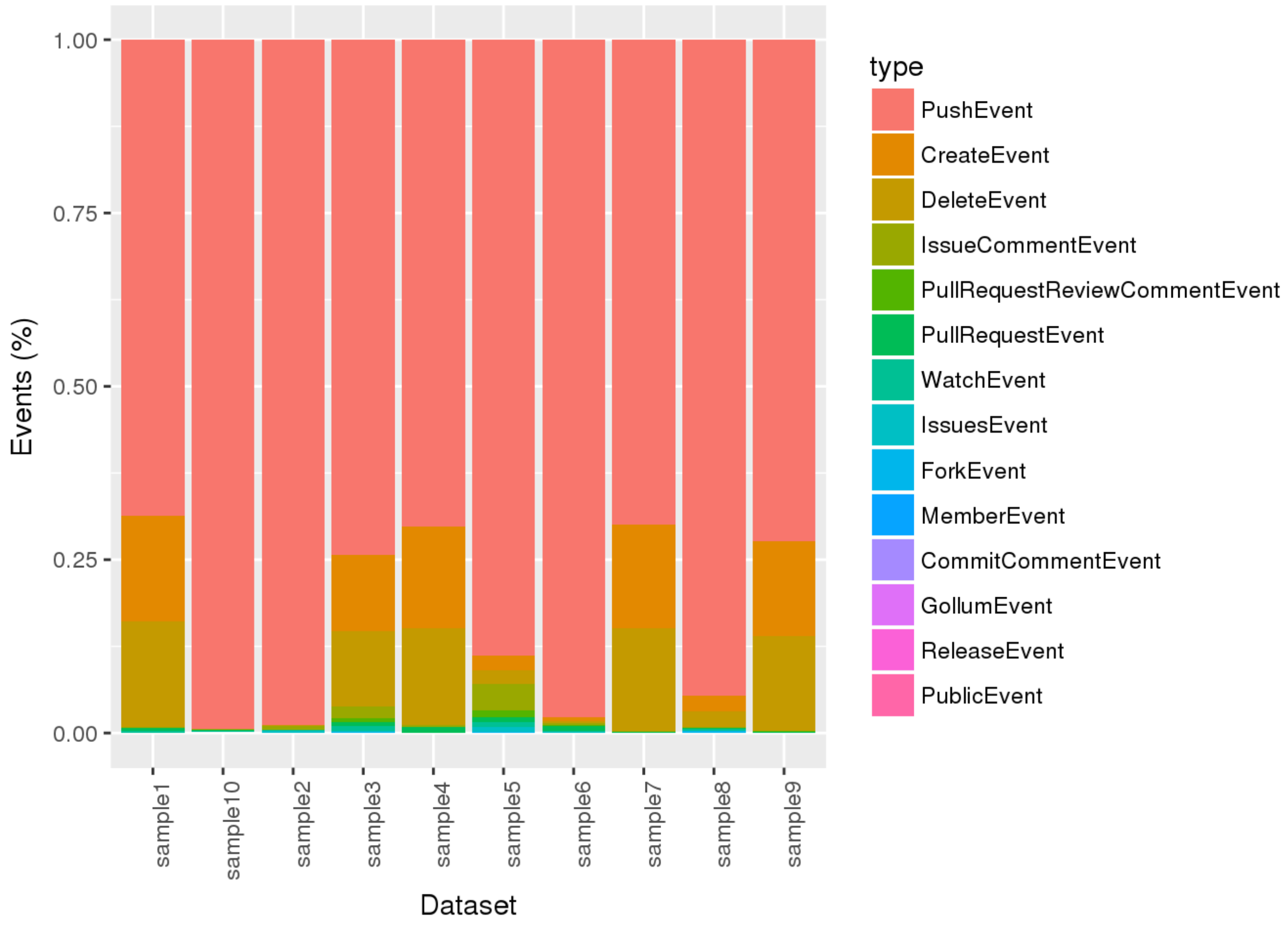
## Event Type Frequency

The distribution of events per event type per sample dataset.

```
push_types_summary <- merge(push_dataset_summary, push_events_repo_samples, by="dataset")

push_sample_event_types <- push_types_summary %>%
  group_by(dataset, type) %>%
  summarise(
    events_sum = sum(num_events),
    total_events_sum = sum(num_repo_events),
    events_prop = events_sum/min(events_in_dataset))

push_sample_event_types$type <- factor(
  push_sample_event_types$type,
  levels = unique(push_sample_event_types$type[order(push_sample_event_types$events_prop, decreasing=TRUE)]))

ggplot(data = push_sample_event_types,
       aes(x = dataset,
           y = events_sum,
           fill=type)) +
  geom_bar(stat="identity", position="stack") +
  ylab("Events") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



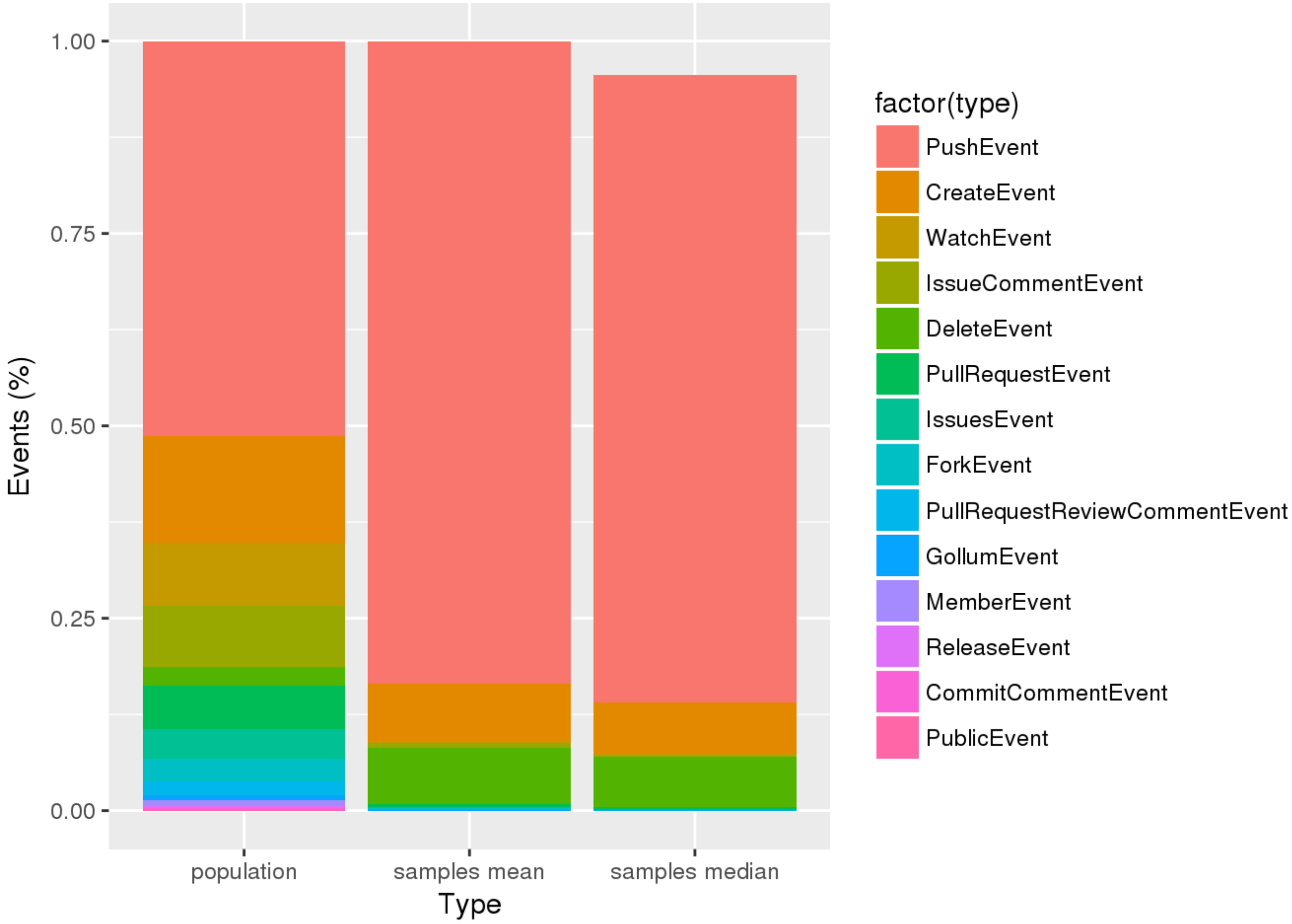
```
ggplot(data = push_sample_event_types,
       aes(x = dataset,
           y = events_prop,
           fill=type)) +
  geom_bar(stat="identity", position="stack") +
  ylab("Events (%)") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```r
participation_rate_event_types <- readRDS("participation_rate_event_types.rds")

push_vs_pop_et_pop <- participation_rate_event_types %>%
  group_by(type) %>%
  summarise(
    dataset = "population",
    events_sum = sum(num_events),
    events_prop = events_sum/total_events
  ) %>%
  select(dataset,
    type,
    events_prop)

push_vs_pop_et_push_mean <- push_sample_event_types %>%
  group_by(type) %>%
  summarise( dataset = "samples mean",
    events_prop = mean(events_prop))

push_vs_pop_et_push_med <- push_sample_event_types %>%
  group_by(type) %>%
  summarise( dataset = "samples median",
    events_prop = median(events_prop))

push_vs_pop_types <- bind_rows(push_vs_pop_et_push_mean, push_vs_pop_et_push_med, push
_vs_pop_et_pop)
```

```
## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector
```

```r
push_vs_pop_types$type <- factor(
  push_vs_pop_types$type,
  levels = unique(push_vs_pop_types$type[order(push_vs_pop_types$events_prop, decreasi
ng=TRUE)]))

ggplot(data = push_vs_pop_types,
       aes(x=dataset, y=events_prop, fill=factor(type))) +
  geom_bar(stat="identity", position="stack") +
  xlab("Type") +
  ylab("Events (%)")
```

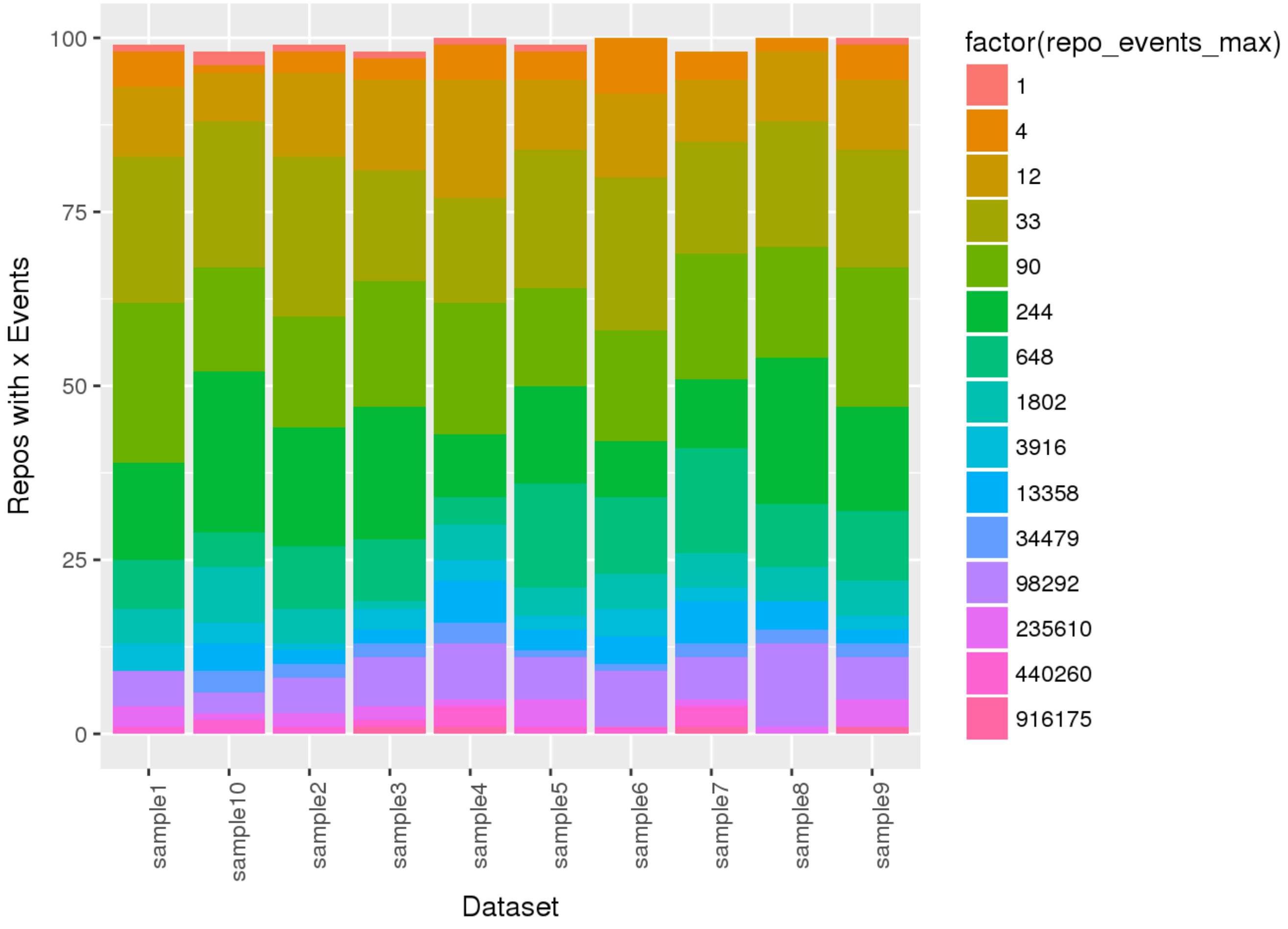


```
ggsave(filename="push_vs_pop_types.png")
```

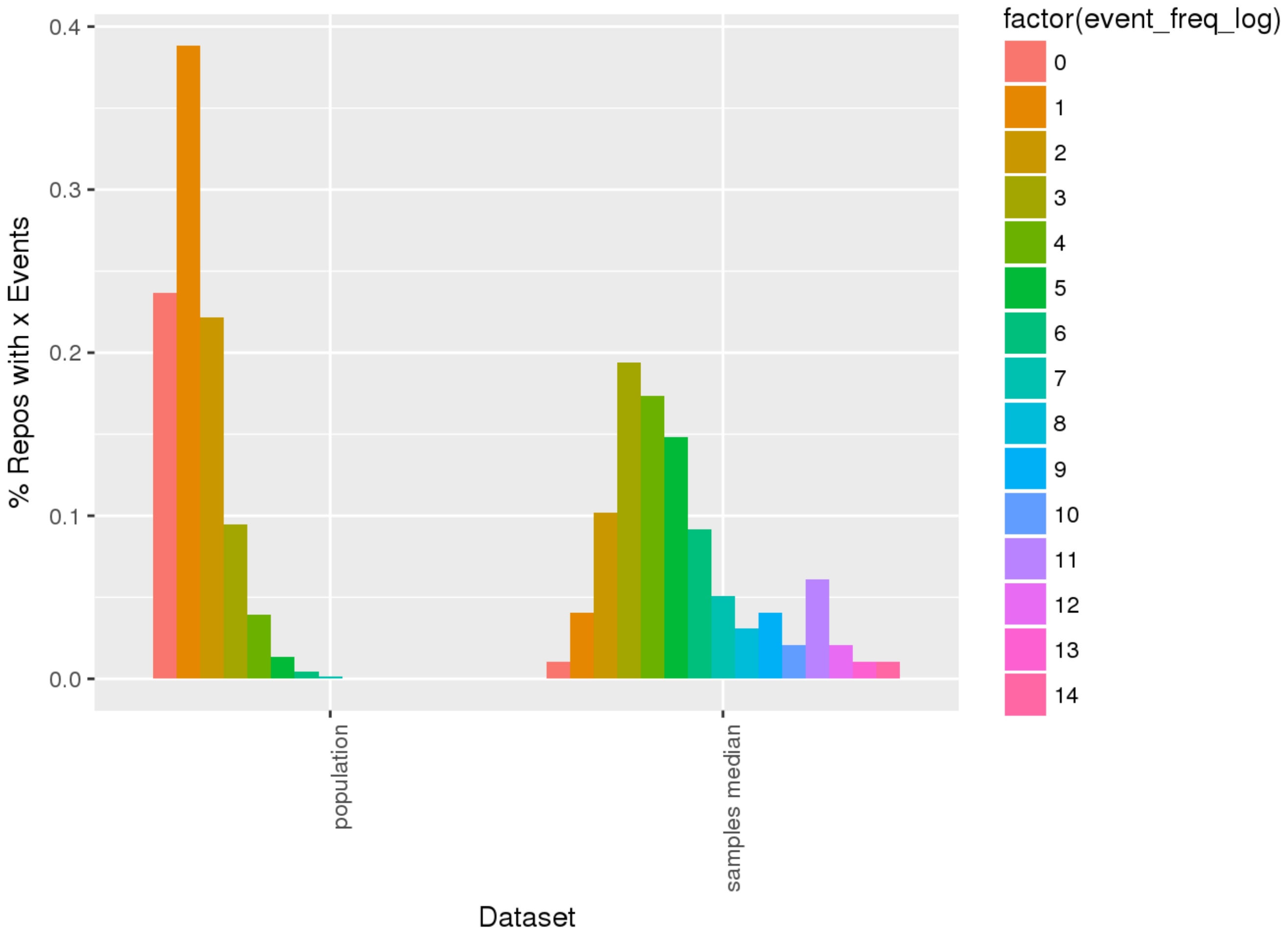
```
## Saving 7 x 5 in image
```

## Events Per Repo

```
push_events <- push_repo_summary %>%
  group_by(dataset, repo_events_log) %>%
  summarise(repo_count = n(),
            num_repo_events_min = min(num_repo_events),
            num_repo_events_max = max(num_repo_events)) %>%
  select(dataset, repo_events_log, repo_count, num_repo_events_max, num_repo_events_mi
n)

push_events_log <- push_events %>%
  group_by(repo_events_log) %>%
  summarise(repo_events_max = max(num_repo_events_max),
            repo_events_min = min(num_repo_events_min))

push_events <- merge(push_events, push_events_log, by="repo_events_log")

push_events <- push_events[order(push_events$repo_count, decreasing=TRUE),]

ggplot(data = push_events,
       aes(x = dataset,
           y = repo_count,
           fill=factor(repo_events_max))) +
  geom_bar(stat="identity", position="stack") +
  ylab("Repos with x Events") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



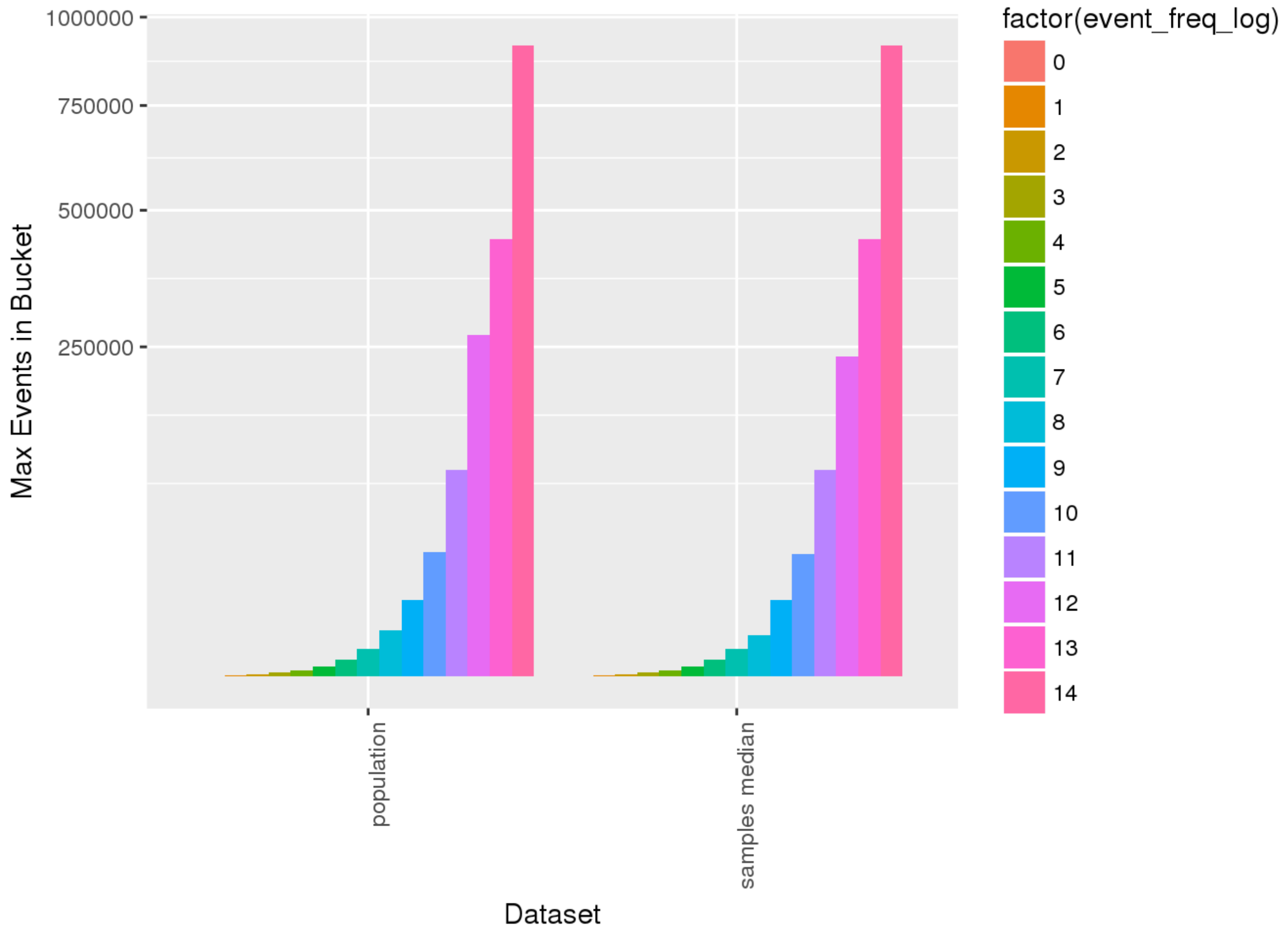
```r
participation_num_events_freq <- readRDS("participation_num_events_freq.rds")
push_events_summary <- merge(push_dataset_summary, push_events, by="dataset")

push_vs_pop_events_pop <- participation_num_events_freq %>%
  group_by(event_freq_log) %>%
  summarise(repo_count = sum(num_repos),
            repo_prop = repo_count/total_repos,
            repo_events_max = max(event_freq),
            dataset = "population")

push_vs_pop_events_push_med <- push_events_summary %>%
  mutate(event_freq_log = repo_events_log) %>%
  group_by(event_freq_log) %>%
  summarise(dataset = "samples median",
    repo_count = median(repo_count),
    repo_prop = repo_count/min(repos_in_dataset),
    repo_events_max = median(repo_events_max))

push_vs_pop_events <- bind_rows(push_vs_pop_events_pop, push_vs_pop_events_push_med)

# push_vs_pop_events$type <- factor(
#   push_vs_pop_types$type,
#   levels = unique(push_vs_pop_types$type[order(push_vs_pop_types$events_prop, decreasing=TRUE)]))

ggplot(data = push_vs_pop_events,
       aes(x = dataset,
           y = repo_prop,
           fill=factor(event_freq_log))) +
  geom_bar(stat="identity", position="dodge") +
  ylab("% Repos with x Events") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
ggsave(filename="push_vs_pop_events.png")
```

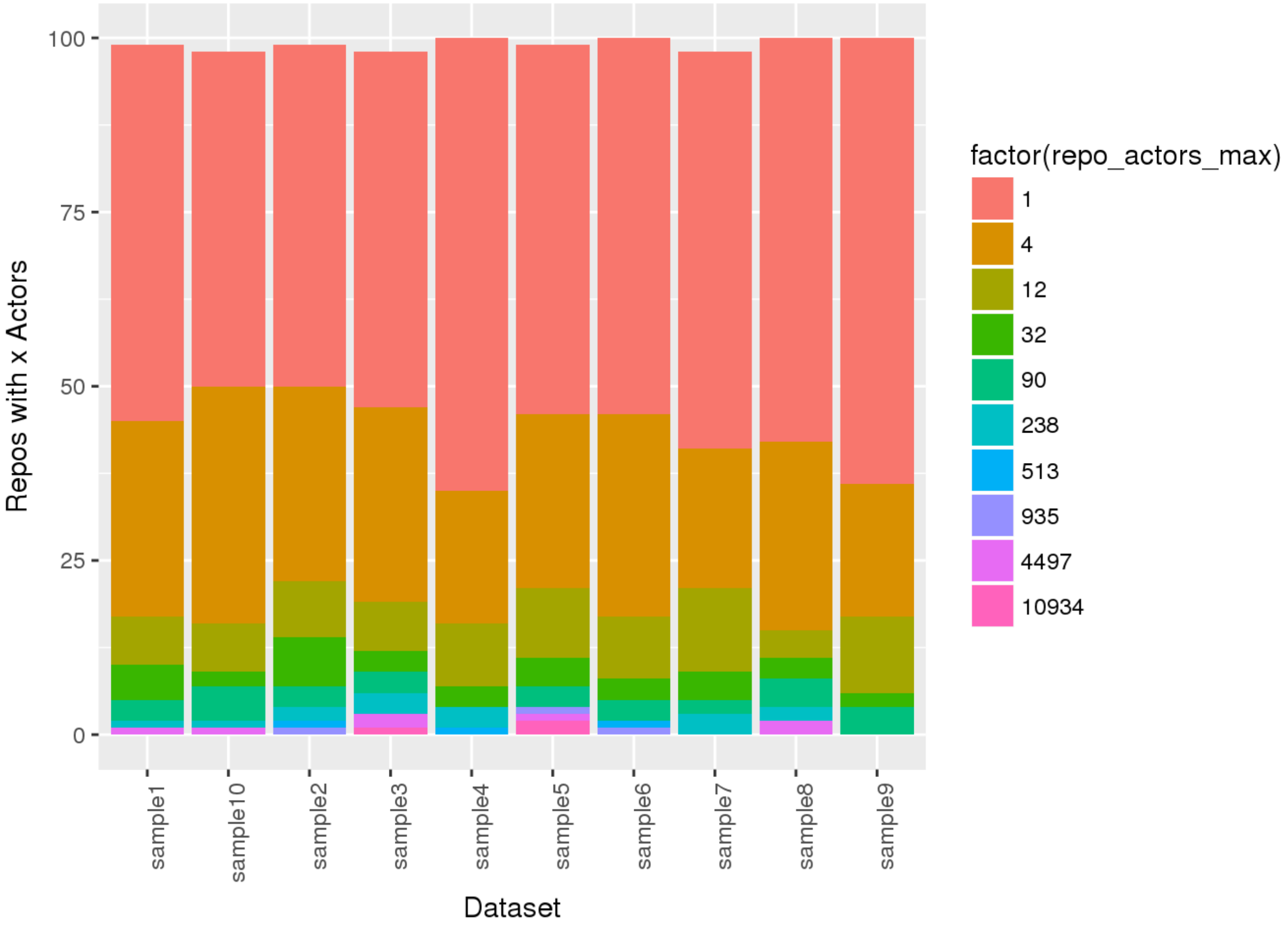
```
## Saving 7 x 5 in image
```

```
ggplot(data = push_vs_pop_events,
       aes(x = dataset,
           y = repo_events_max,
           fill=factor(event_freq_log))) +
  geom_bar(stat="identity", position="dodge") +
  ylab("Max Events in Bucket") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_y_continuous(trans="sqrt")
```

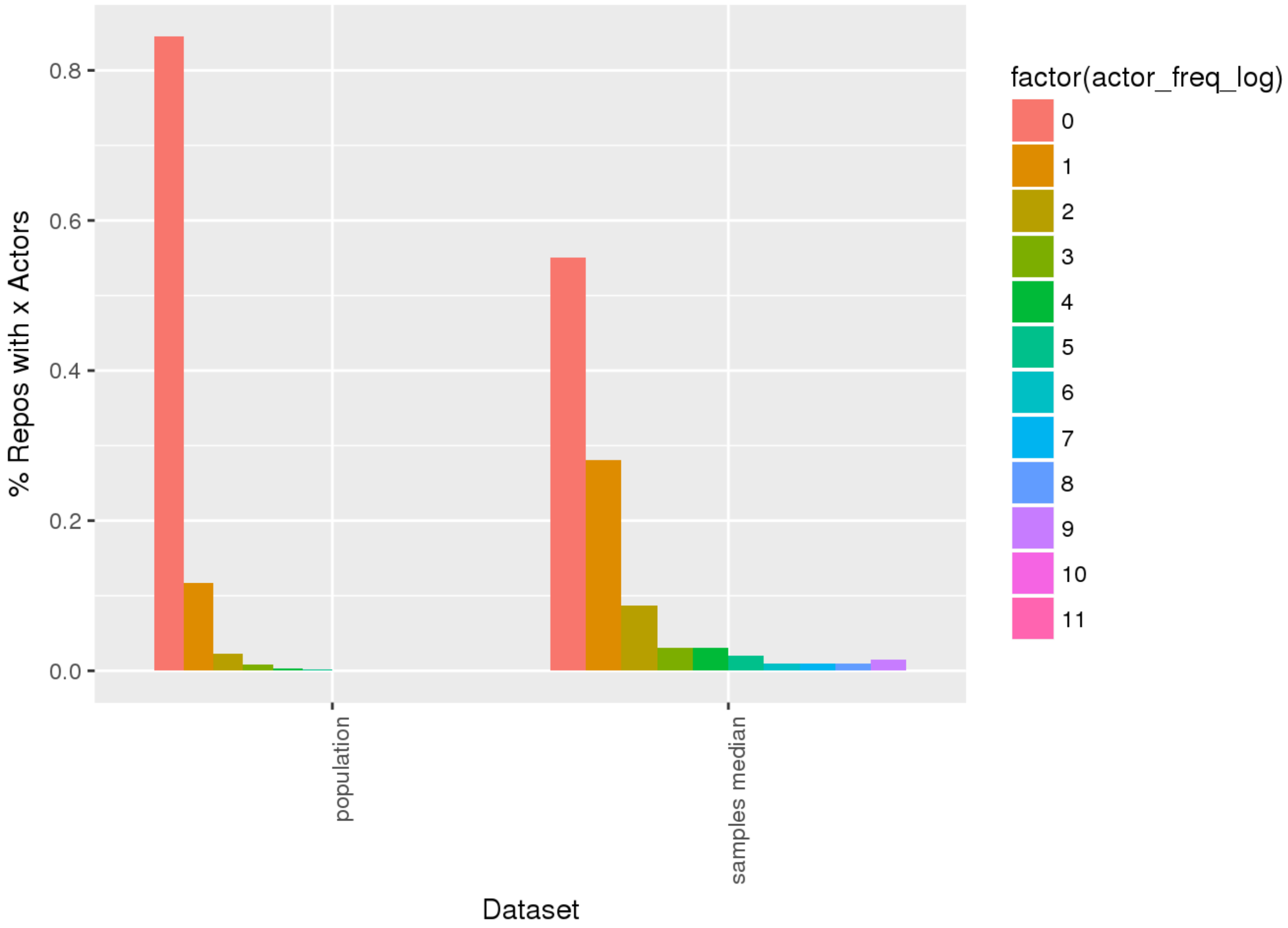


# Actors Per Repo

```
push_actors <- push_repo_summary %>%
  group_by(dataset, repo_actors_log) %>%
  summarise(repo_count = n(),
            num_repo_actors_min = min(num_repo_actors),
            num_repo_actors_max = max(num_repo_actors)) %>%
  select(dataset, repo_actors_log, repo_count, num_repo_actors_max, num_repo_actors_mi
n)

push_actors_log <- push_actors %>%
  group_by(repo_actors_log) %>%
  summarise(repo_actors_max = max(num_repo_actors_max),
            repo_actors_min = min(num_repo_actors_min))

push_actors <- merge(push_actors, push_actors_log, by="repo_actors_log")

push_actors <- push_actors[order(push_actors$repo_count, decreasing=TRUE),]

ggplot(data = push_actors,
       aes(x = dataset,
           y = repo_count,
           fill=factor(repo_actors_max))) +
  geom_bar(stat="identity", position="stack") +
  ylab("Repos with x Actors") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



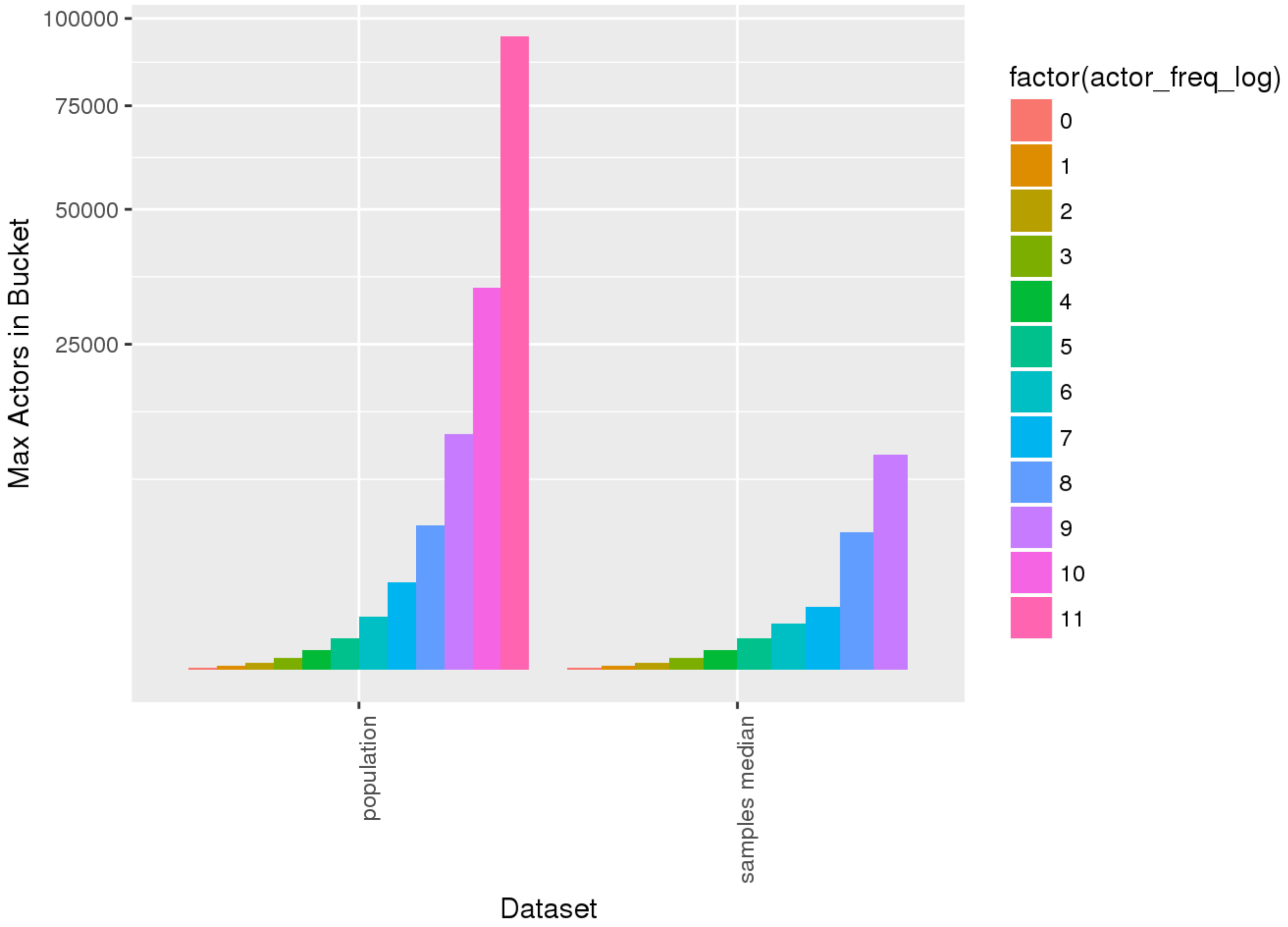
```r
# this needs to be proportions

participation_num_actors_freq <- readRDS("participation_num_actors_freq.rds")
push_actors_summary <- merge(push_dataset_summary, push_actors, by="dataset")

push_vs_pop_actors_pop <- participation_num_actors_freq %>%
  group_by(actor_freq_log) %>%
  summarise(repo_count = sum(num_repos),
            repo_prop = repo_count/total_repos,
            repo_actors_max = max(actor_freq),
            dataset = "population")

push_vs_pop_actors_push_med <- push_actors_summary %>%
  mutate(actor_freq_log = repo_actors_log) %>%
  group_by(actor_freq_log) %>%
  summarise(dataset = "samples median",
    repo_count = median(repo_count),
    repo_prop = repo_count/min(repos_in_dataset),
    repo_actors_max = median(repo_actors_max))

push_vs_pop_actors <- bind_rows(push_vs_pop_actors_pop, push_vs_pop_actors_push_med)

# push_vs_pop_events$type <- factor(
#   push_vs_pop_types$type,
#   levels = unique(push_vs_pop_types$type[order(push_vs_pop_types$events_prop, decrea
sing=TRUE)]))

ggplot(data = push_vs_pop_actors,
       aes(x = dataset,
           y = repo_prop,
           fill=factor(actor_freq_log))) +
  geom_bar(stat="identity", position="dodge") +
  ylab("% Repos with x Actors") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
ggsave(filename="push_vs_pop_actors.png")
```

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

```
ggplot(data = push_vs_pop_actors,
       aes(x = dataset,
           y = repo_actors_max,
           fill=factor(actor_freq_log))) +
  geom_bar(stat="identity", position="dodge") +
  ylab("Max Actors in Bucket") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_y_continuous(trans="sqrt")
```



# High Participation

Watch, Issue Comment, and Fork events occur most frequently with high participation repositories therefore samples taken from these events would have a higher probability of containing this type of repository.

```
watch_events_repo_samples <- readRDS("watch_events_repo_samples.rds")

watch_events_repo_samples <-
    mutate(watch_events_repo_samples,
           participation_level = ifelse(participation_rate < 1 & participation_rate >
0, 'Medium', ''))

watch_events_repo_samples <-
  mutate(watch_events_repo_samples, participation_level = ifelse(participation_rate ==
 0, 'High', participation_level))

watch_events_repo_samples <-
  mutate(watch_events_repo_samples, participation_level = ifelse(participation_rate ==
 1, 'Low', participation_level))
```

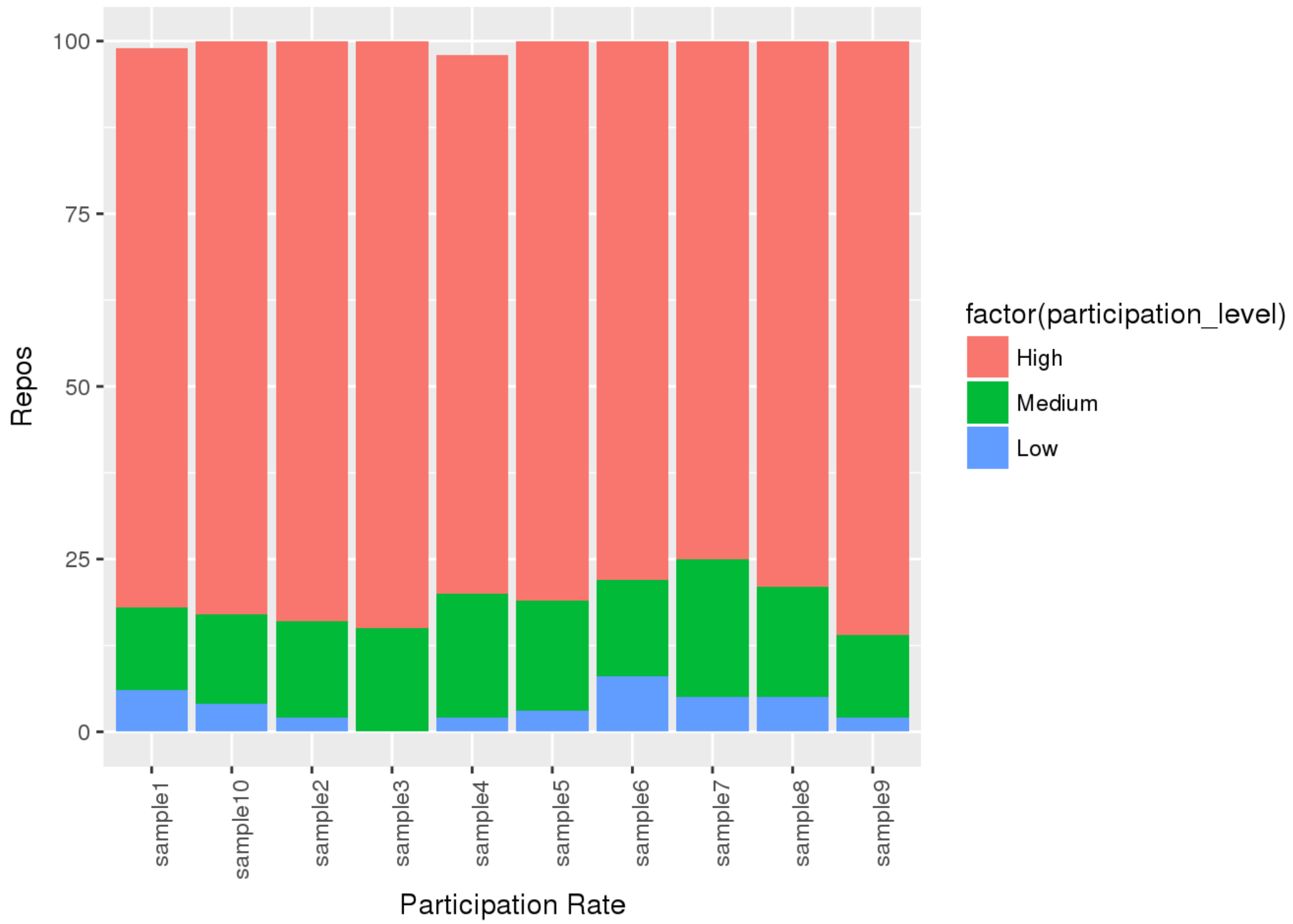
```
isscomment_events_repo_samples <- readRDS("isscomment_events_repo_samples.rds")

isscomment_events_repo_samples <-
    mutate(isscomment_events_repo_samples,
           participation_level = ifelse(participation_rate < 1 & participation_rate >
0, 'Medium', ''))

isscomment_events_repo_samples <-
  mutate(isscomment_events_repo_samples, participation_level = ifelse(participation_ra
te == 0, 'High', participation_level))

isscomment_events_repo_samples <-
  mutate(isscomment_events_repo_samples, participation_level = ifelse(participation_ra
te == 1, 'Low', participation_level))
```

```
fork_events_repo_samples <- readRDS("fork_events_repo_samples.rds")

fork_events_repo_samples <-
    mutate(fork_events_repo_samples,
           participation_level = ifelse(participation_rate < 1 & participation_rate >
0, 'Medium', ''))

fork_events_repo_samples <-
  mutate(fork_events_repo_samples, participation_level = ifelse(participation_rate ==
0, 'High', participation_level))

fork_events_repo_samples <-
  mutate(fork_events_repo_samples, participation_level = ifelse(participation_rate ==
1, 'Low', participation_level))
```

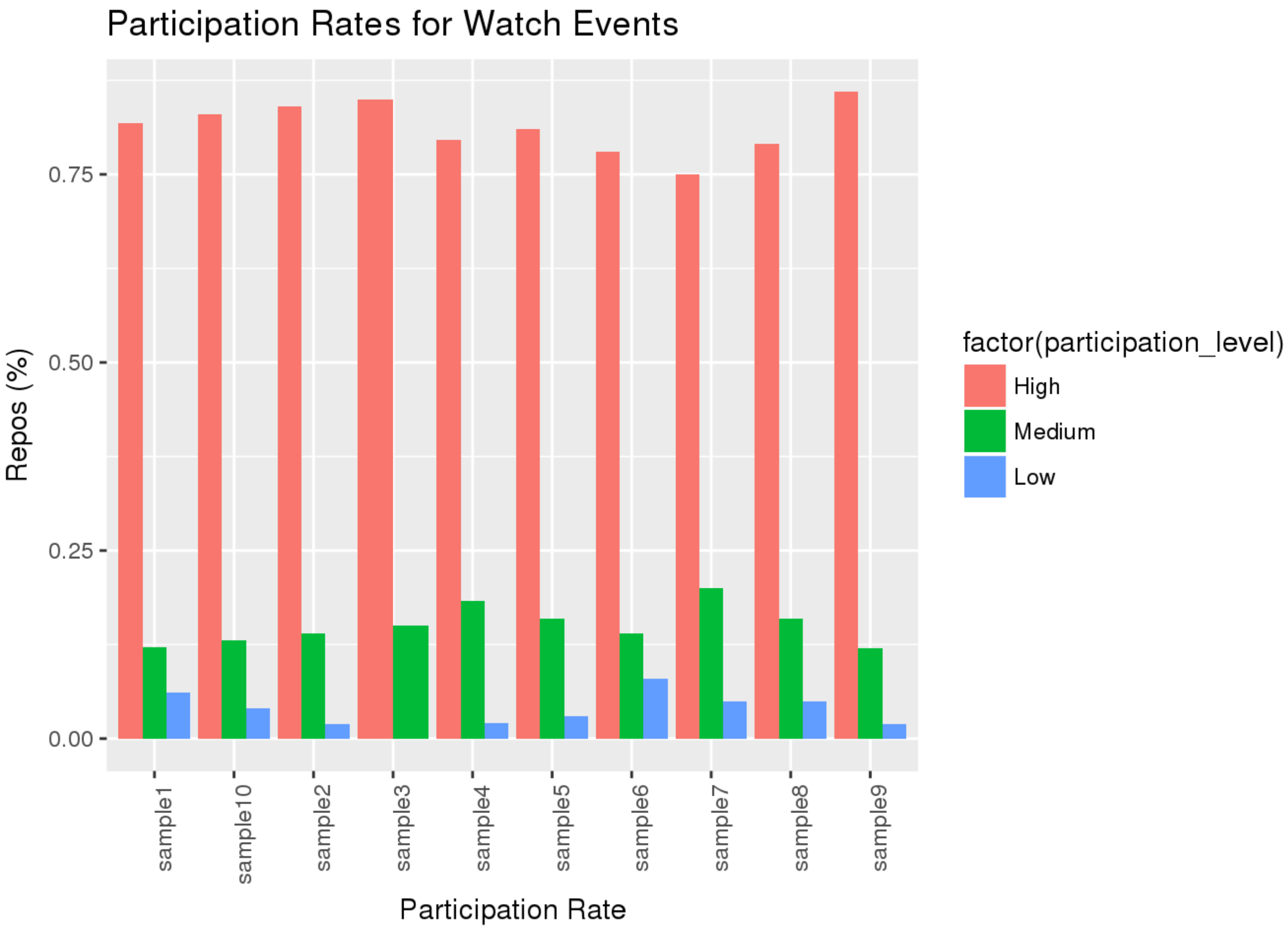
## Participation Rates

First we examine if the samples drawn by event type show a higher proportion of high participation repositories.

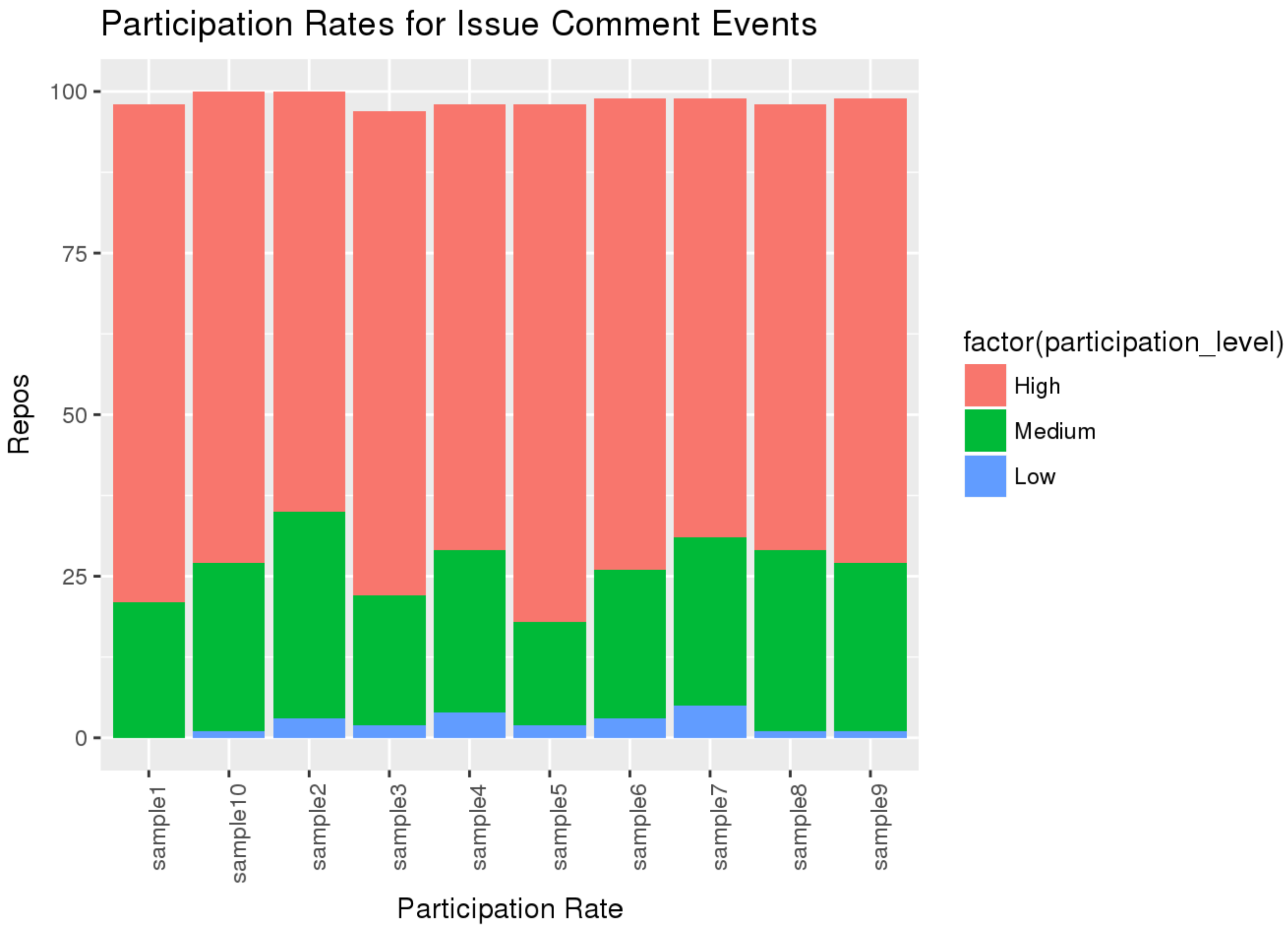
```
watch_repo_summary <- watch_events_repo_samples %>%
  group_by(dataset, repo_name) %>%
  summarise(
    participation_level = max(participation_level),
    num_repo_events = max(num_repo_events),
    num_repo_actors = max(num_repo_actors),
    repo_actors_log = round(log(num_repo_actors)),
    repo_events_log = round(log(num_repo_events))
  )

watch_dataset_summary <- watch_repo_summary %>%
  group_by(dataset) %>%
  summarise(repos_in_dataset = n(),
            actors_in_dataset = sum(num_repo_actors),
            events_in_dataset = sum(num_repo_events))

watch_participation_summary <- watch_repo_summary %>%
  group_by(dataset, participation_level) %>%
  summarise(num_repos = n())

watch_participation_summary <- merge(watch_participation_summary, watch_dataset_summary, by="dataset")

watch_participation_summary <- watch_participation_summary %>%
  mutate(repos_perc = num_repos/repos_in_dataset)

watch_participation_summary$participation_level <-
factor(watch_participation_summary$participation_level,
                           levels = c("High", "Medium", "Low"))

ggplot(data = watch_participation_summary,
       aes(x=dataset, y=num_repos, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="stack") +
  xlab("Participation Rate") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



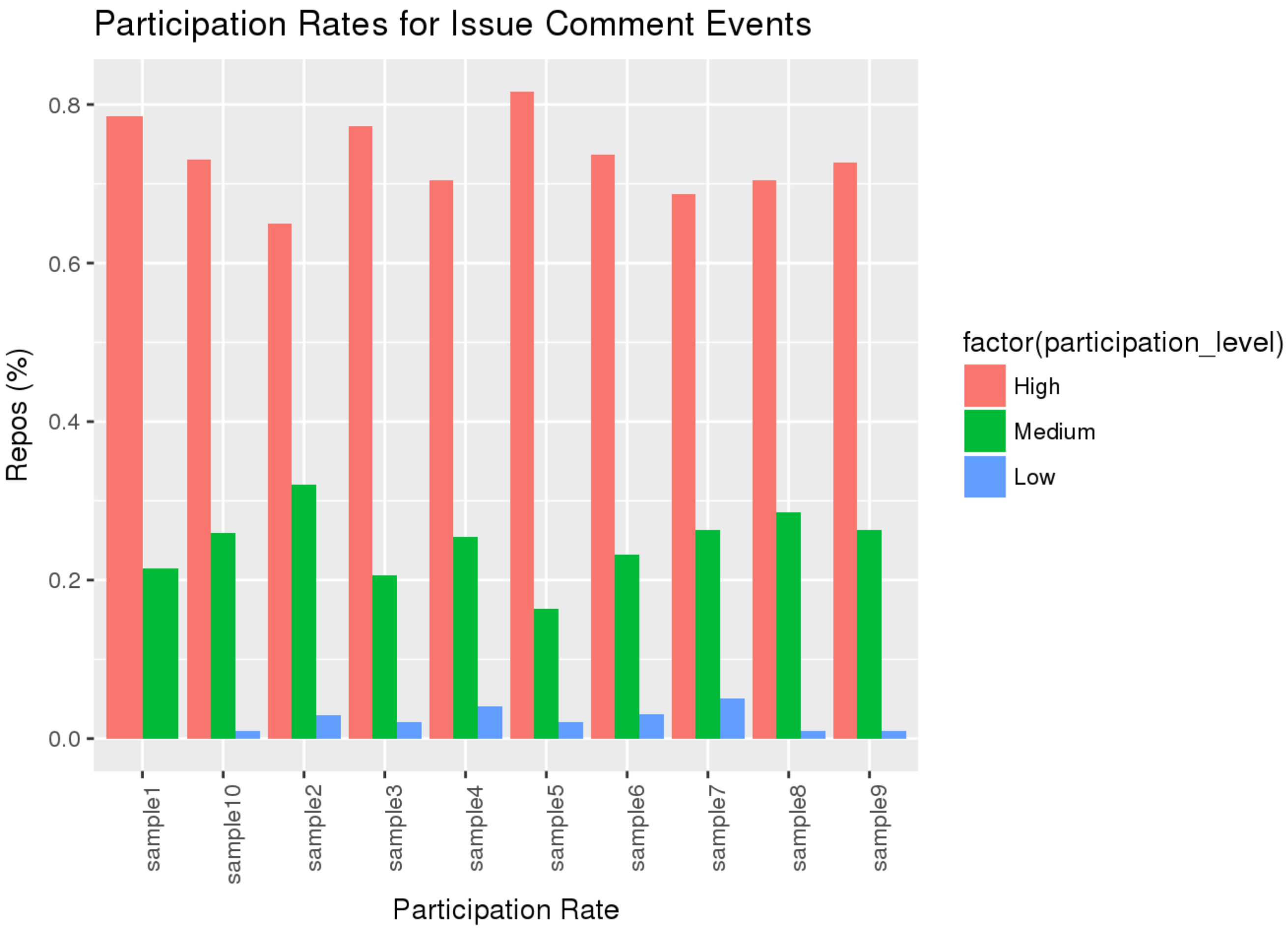
```
ggplot(data = watch_participation_summary,
       aes(x=dataset, y=repos_perc, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Watch Events")
```


Participation Rates for Watch Events
Repos (%)
0.00
0.25
0.50
0.75
sample1
sample10
sample2
sample3
sample4
sample5
sample6
sample7
sample8
sample9
Participation Rate
factor(participation_level)
High
Medium
Low

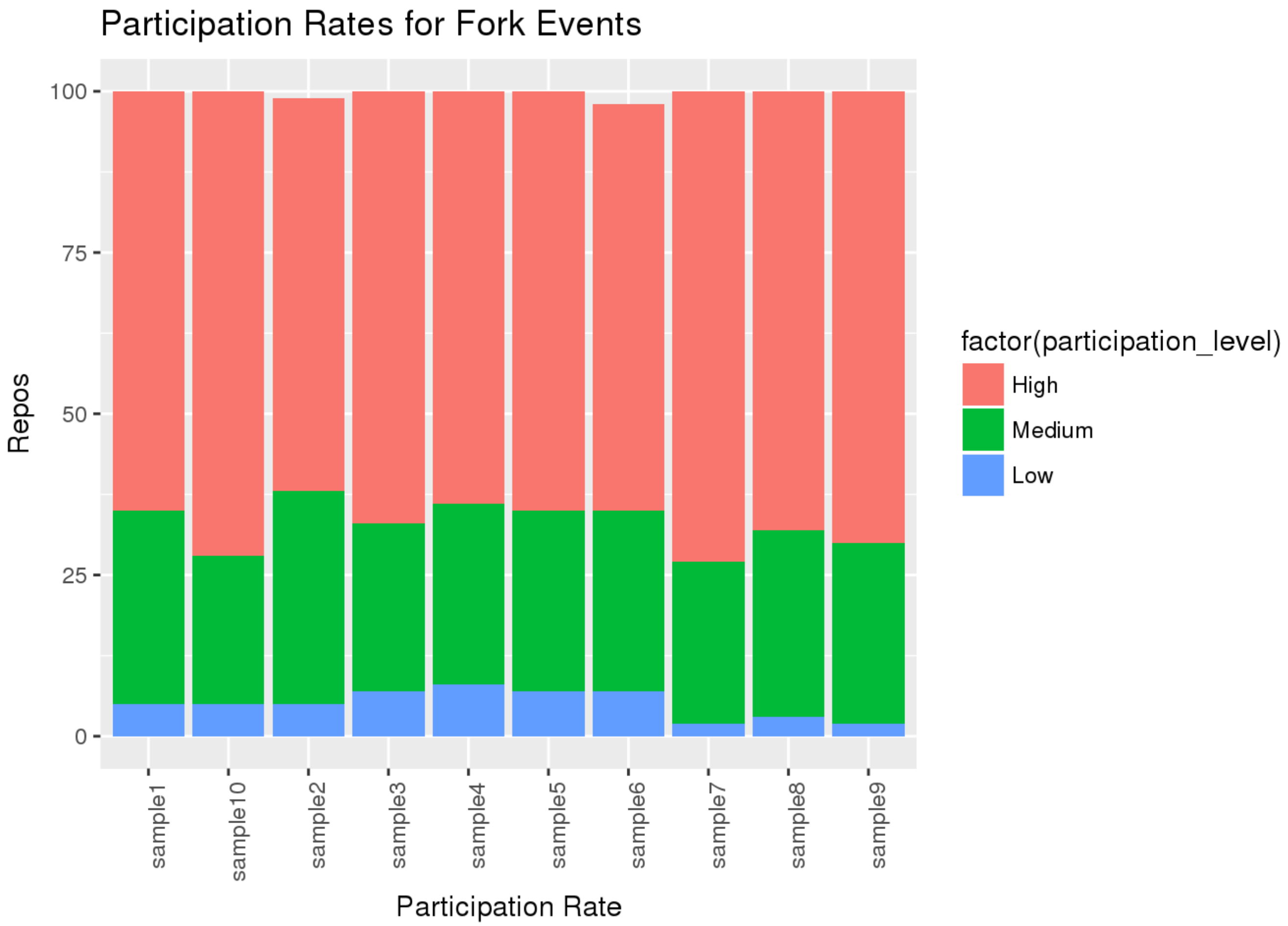
```
isscomment_repo_summary <- isscomment_events_repo_samples %>%
  group_by(dataset, repo_name) %>%
  summarise(
    participation_level = max(participation_level),
    num_repo_events = max(num_repo_events),
    num_repo_actors = max(num_repo_actors),
    repo_actors_log = round(log(num_repo_actors)),
    repo_events_log = round(log(num_repo_events))
  )

isscomment_dataset_summary <- isscomment_repo_summary %>%
  group_by(dataset) %>%
  summarise(repos_in_dataset = n(),
            actors_in_dataset = sum(num_repo_actors),
            events_in_dataset = sum(num_repo_events))

isscomment_participation_summary <- isscomment_repo_summary %>%
  group_by(dataset, participation_level) %>%
  summarise(num_repos = n())

isscomment_participation_summary <- merge(isscomment_participation_summary, isscomment
_dataset_summary, by="dataset")

isscomment_participation_summary <- isscomment_participation_summary %>%
  mutate(repos_perc = num_repos/repos_in_dataset)

isscomment_participation_summary$participation_level <- factor(isscomment_participatio
n_summary$participation_level,
                           levels = c("High", "Medium", "Low"))

ggplot(data = isscomment_participation_summary,
       aes(x=dataset, y=num_repos, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="stack") +
  xlab("Participation Rate") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Issue Comment Events")
```



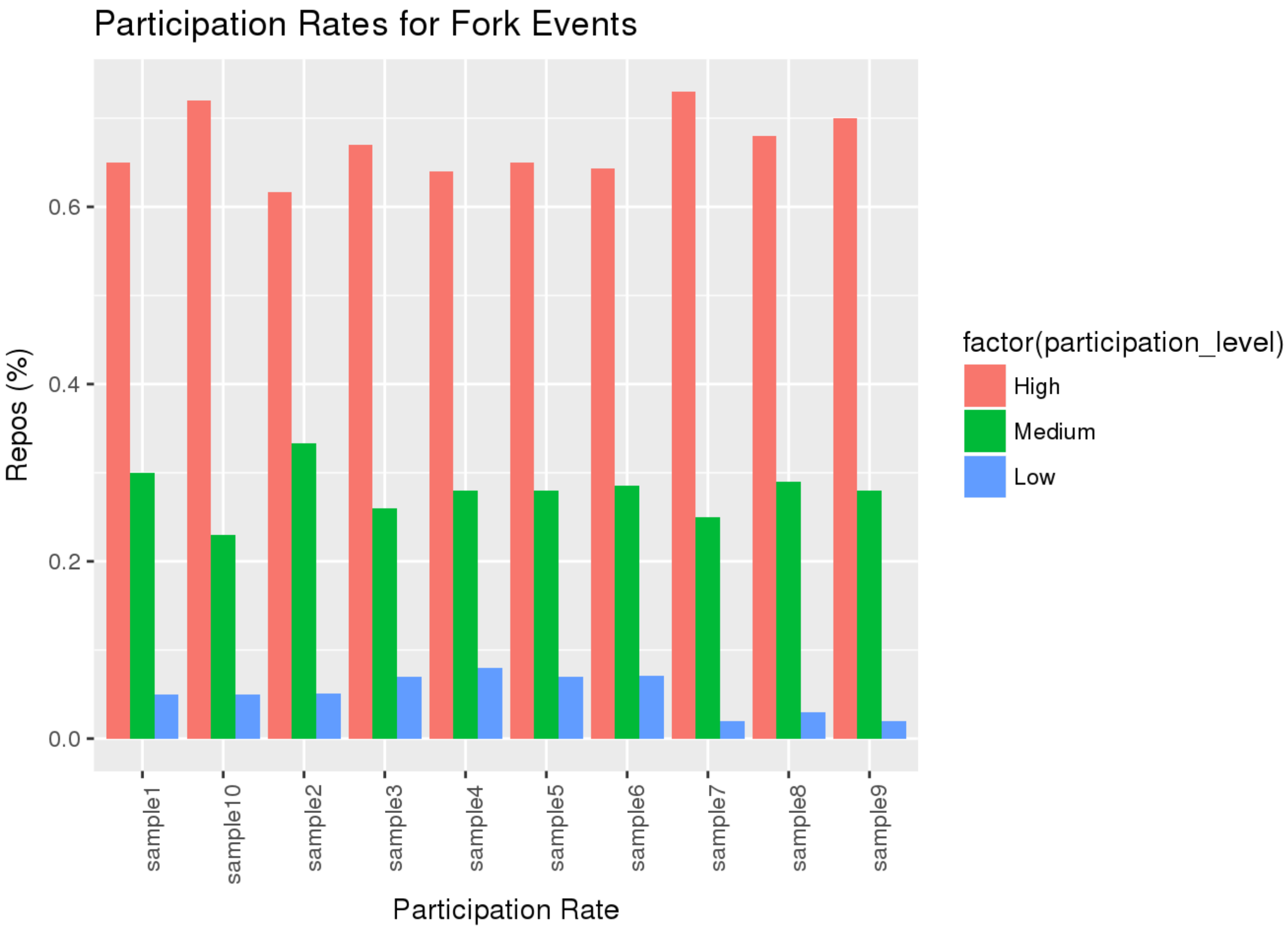
```
ggplot(data = isscomment_participation_summary,
       aes(x=dataset, y=repos_perc, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Issue Comment Events")
```


Participation Rates for Issue Comment Events
Repos (%)
0.0
0.2
0.4
0.6
0.8
sample1
sample10
sample2
sample3
sample4
sample5
sample6
sample7
sample8
sample9
Participation Rate
factor(participation_level)
High
Medium
Low

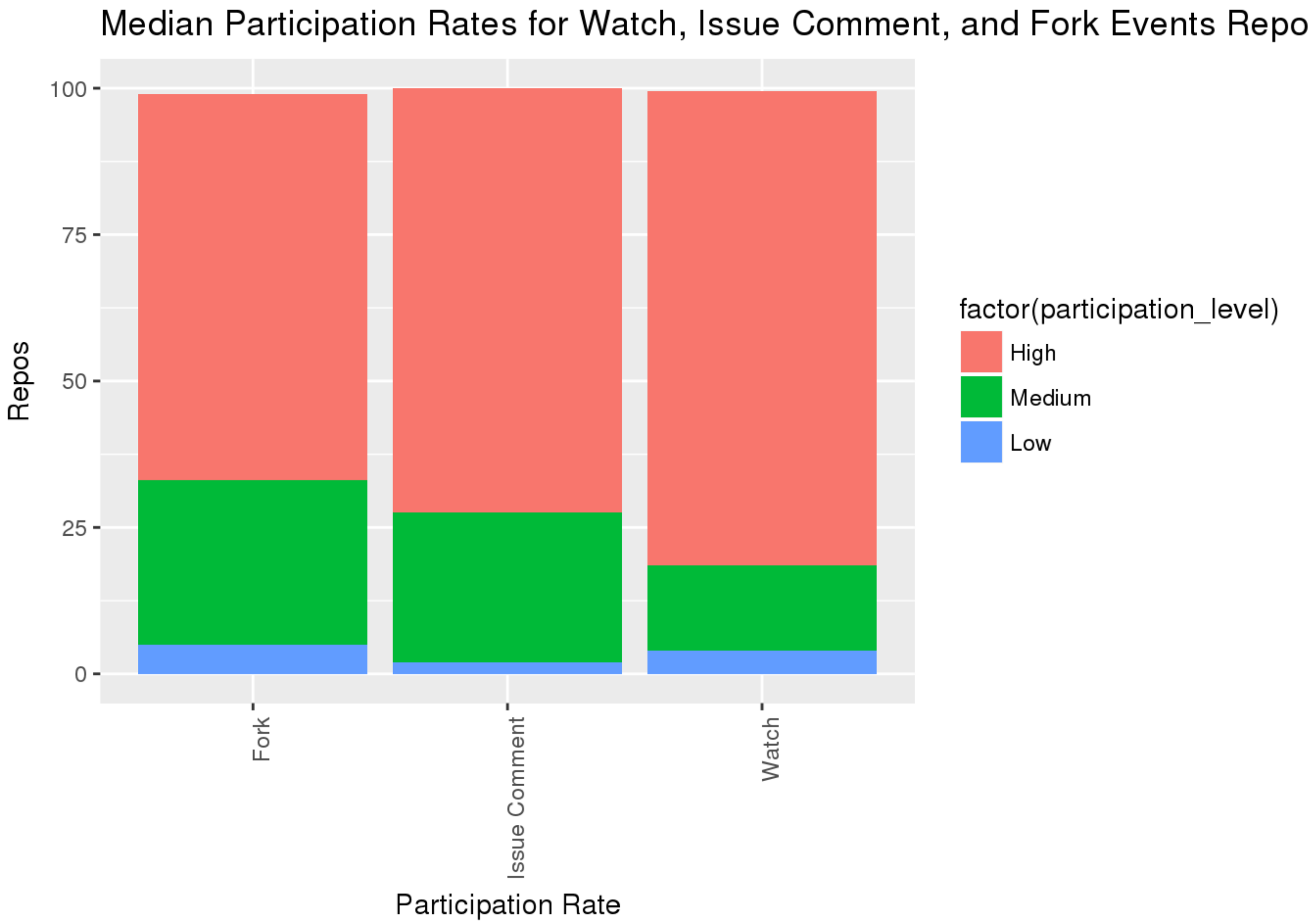
```
fork_repo_summary <- fork_events_repo_samples %>%
  group_by(dataset, repo_name) %>%
  summarise(
    participation_level = max(participation_level),
    num_repo_events = max(num_repo_events),
    num_repo_actors = max(num_repo_actors),
    repo_actors_log = round(log(num_repo_actors)),
    repo_events_log = round(log(num_repo_events))
  )

fork_dataset_summary <- fork_repo_summary %>%
  group_by(dataset) %>%
  summarise(repos_in_dataset = n(),
            actors_in_dataset = sum(num_repo_actors),
            events_in_dataset = sum(num_repo_events))

fork_participation_summary <- fork_repo_summary %>%
  group_by(dataset, participation_level) %>%
  summarise(num_repos = n())

fork_participation_summary <- merge(fork_participation_summary, fork_dataset_summary,
by="dataset")

fork_participation_summary <- fork_participation_summary %>%
  mutate(repos_perc = num_repos/repos_in_dataset)

fork_participation_summary$participation_level <- factor(fork_participation_summary$pa
rticipation_level,
                           levels = c("High", "Medium", "Low"))

ggplot(data = fork_participation_summary,
       aes(x=dataset, y=num_repos, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="stack") +
  xlab("Participation Rate") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Fork Events")
```



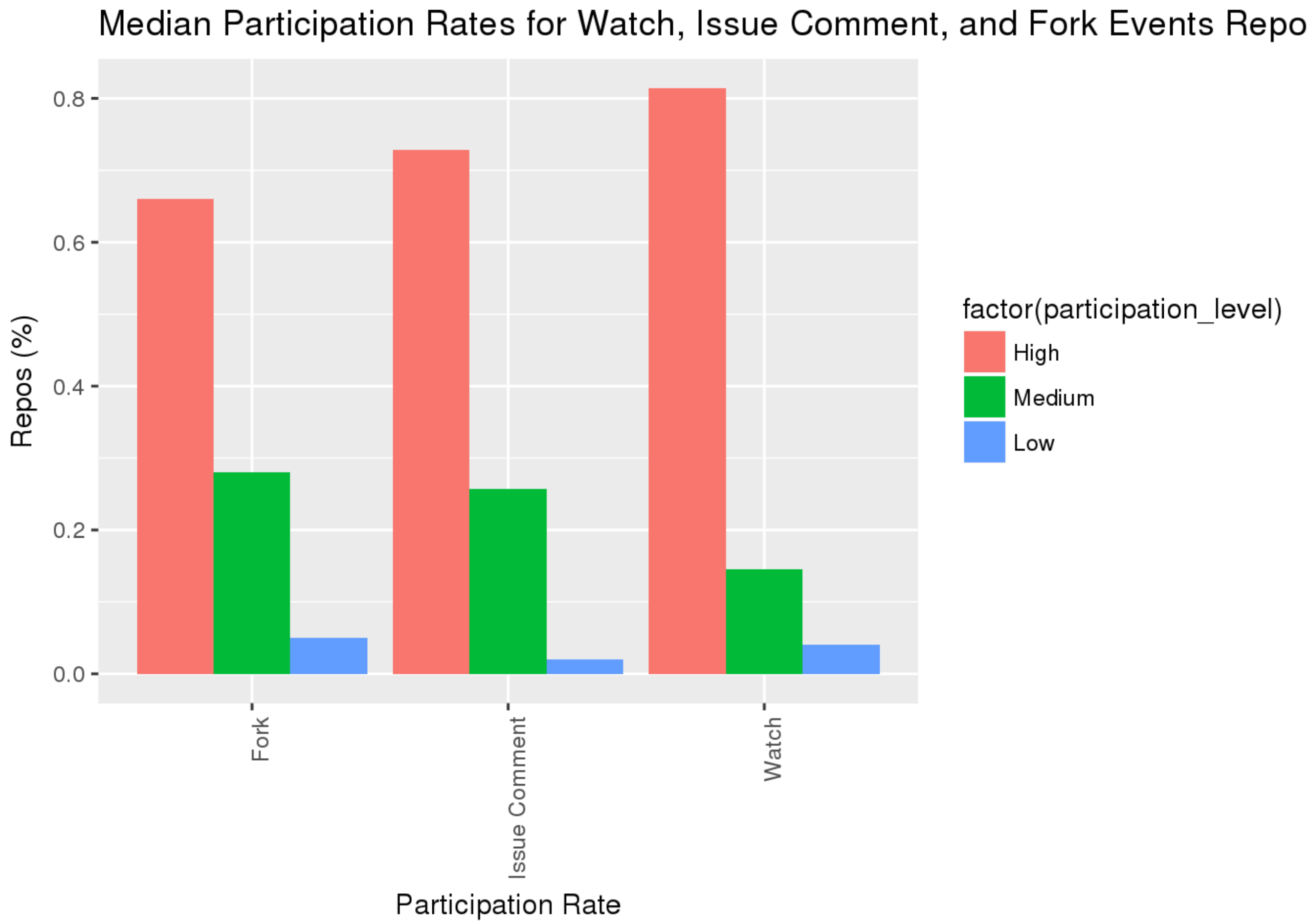
```
ggplot(data = fork_participation_summary,
       aes(x=dataset, y=repos_perc, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Fork Events")
```


Participation Rates for Fork Events
Repos (%)
0.0
0.2
0.4
0.6
sample1
sample10
sample2
sample3
sample4
sample5
sample6
sample7
sample8
sample9
Participation Rate
factor(participation_level)
High
Medium
Low

```
watch_pr <- watch_participation_summary %>%
  group_by(participation_level) %>%
  summarise(num_repos_med = median(num_repos),
            num_repos_mean = mean(num_repos),
            repos_perc_med = median(repos_perc),
            repos_perc_mean = mean(repos_perc),
            dataset = "Watch")

isscomment_pr <- isscomment_participation_summary %>%
  group_by(participation_level) %>%
  summarise(num_repos_med = median(num_repos),
            num_repos_mean = mean(num_repos),
            repos_perc_med = median(repos_perc),
            repos_perc_mean = mean(repos_perc),
            dataset = "Issue Comment")

fork_pr <- fork_participation_summary %>%
  group_by(participation_level) %>%
  summarise(num_repos_med = median(num_repos),
            num_repos_mean = mean(num_repos),
            repos_perc_med = median(repos_perc),
            repos_perc_mean = mean(repos_perc),
            dataset = "Fork")

high_type_samples_pr <- bind_rows(watch_pr, isscomment_pr, fork_pr)

ggplot(data = high_type_samples_pr,
       aes(x=dataset, y=num_repos_med, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="stack") +
  xlab("Participation Rate") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Median Participation Rates for Watch, Issue Comment, and Fork Events Repo S
amples")
```



```
ggplot(data = high_type_samples_pr,
       aes(x=dataset, y=repos_perc_med, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Median Participation Rates for Watch, Issue Comment, and Fork Events Repo S
amples")
```



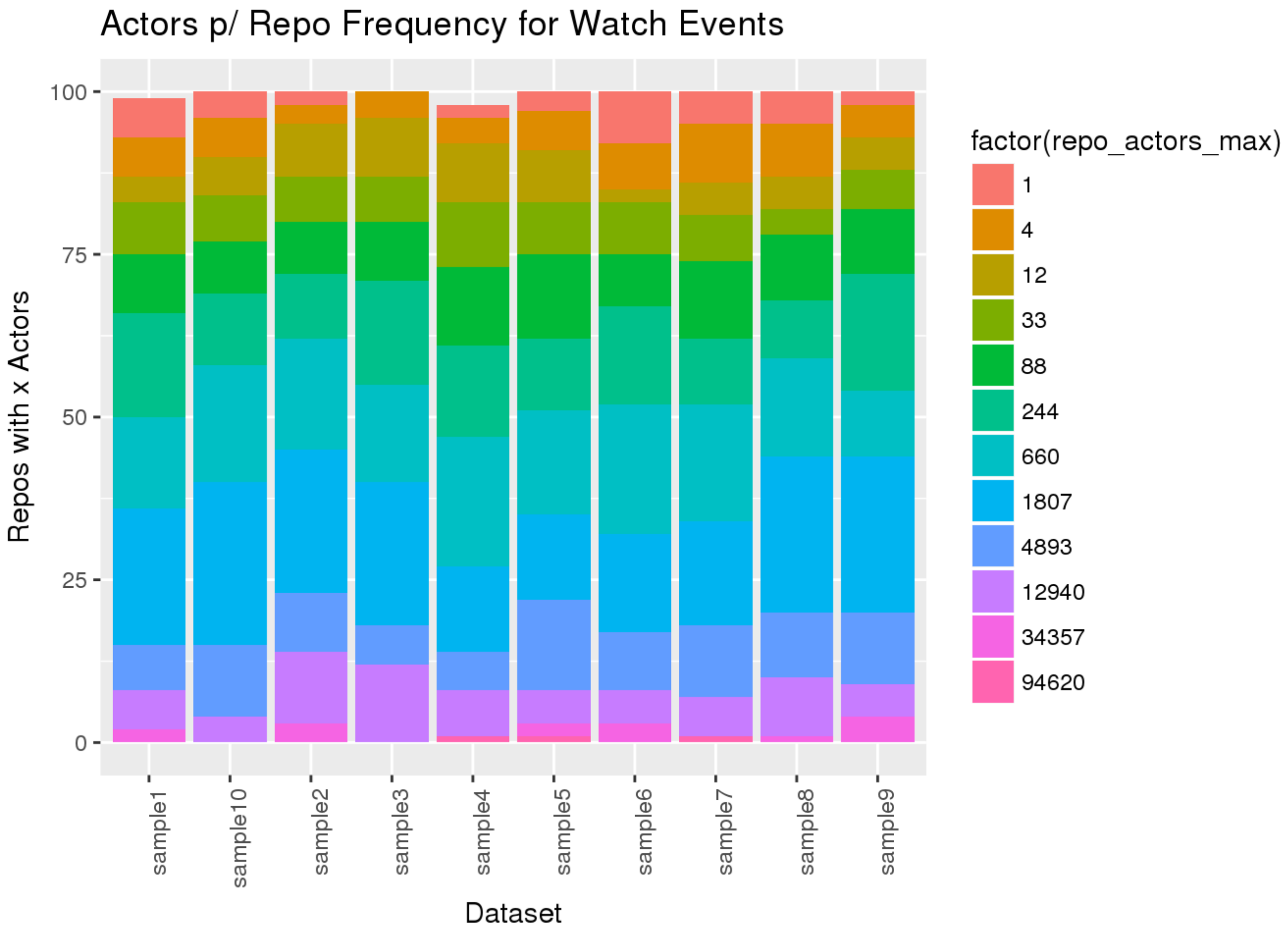
```
ggsave(filename="high_type_samples_participation.png")
```

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

## Actors Per Repo

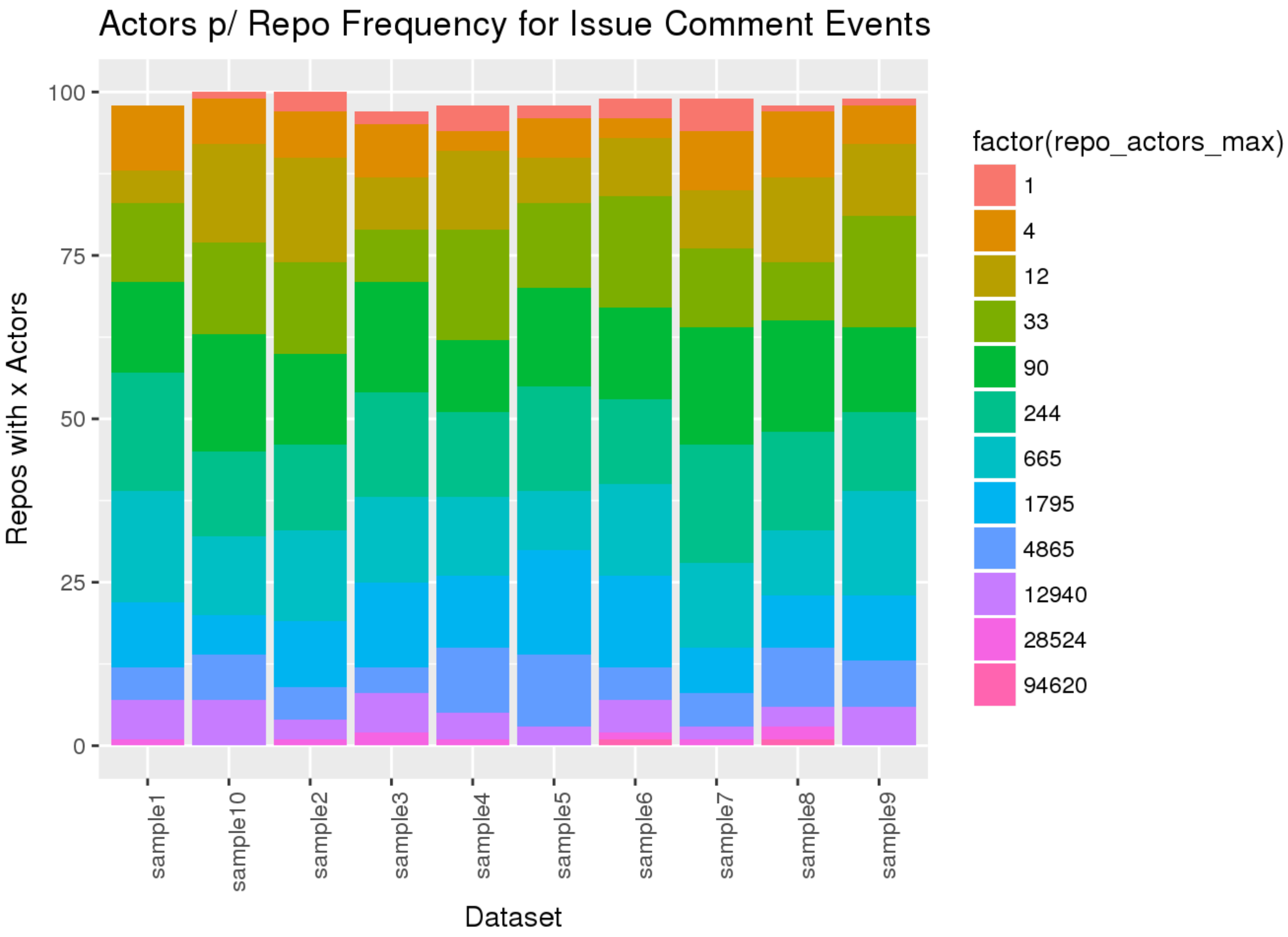
Second, we look at the number of unique actors that generated events and compare that to the high participation population studied earlier.

```r
watch_actors <- watch_repo_summary %>%
  group_by(dataset, repo_actors_log) %>%
  summarise(repo_count = n(),
            num_repo_actors_min = min(num_repo_actors),
            num_repo_actors_max = max(num_repo_actors)) %>%
  select(dataset, repo_actors_log, repo_count, num_repo_actors_max, num_repo_actors_mi
n)

watch_actors_log <- watch_actors %>%
  group_by(repo_actors_log) %>%
  summarise(repo_actors_max = max(num_repo_actors_max),
            repo_actors_min = min(num_repo_actors_min))

watch_actors <- merge(watch_actors, watch_actors_log, by="repo_actors_log")

watch_actors <- watch_actors[order(watch_actors$repo_count, decreasing=TRUE),]

ggplot(data = watch_actors,
       aes(x = dataset,
           y = repo_count,
           fill=factor(repo_actors_max))) +
  geom_bar(stat="identity", position="stack") +
  ylab("Repos with x Actors") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Actors p/ Repo Frequency for Watch Events")
```

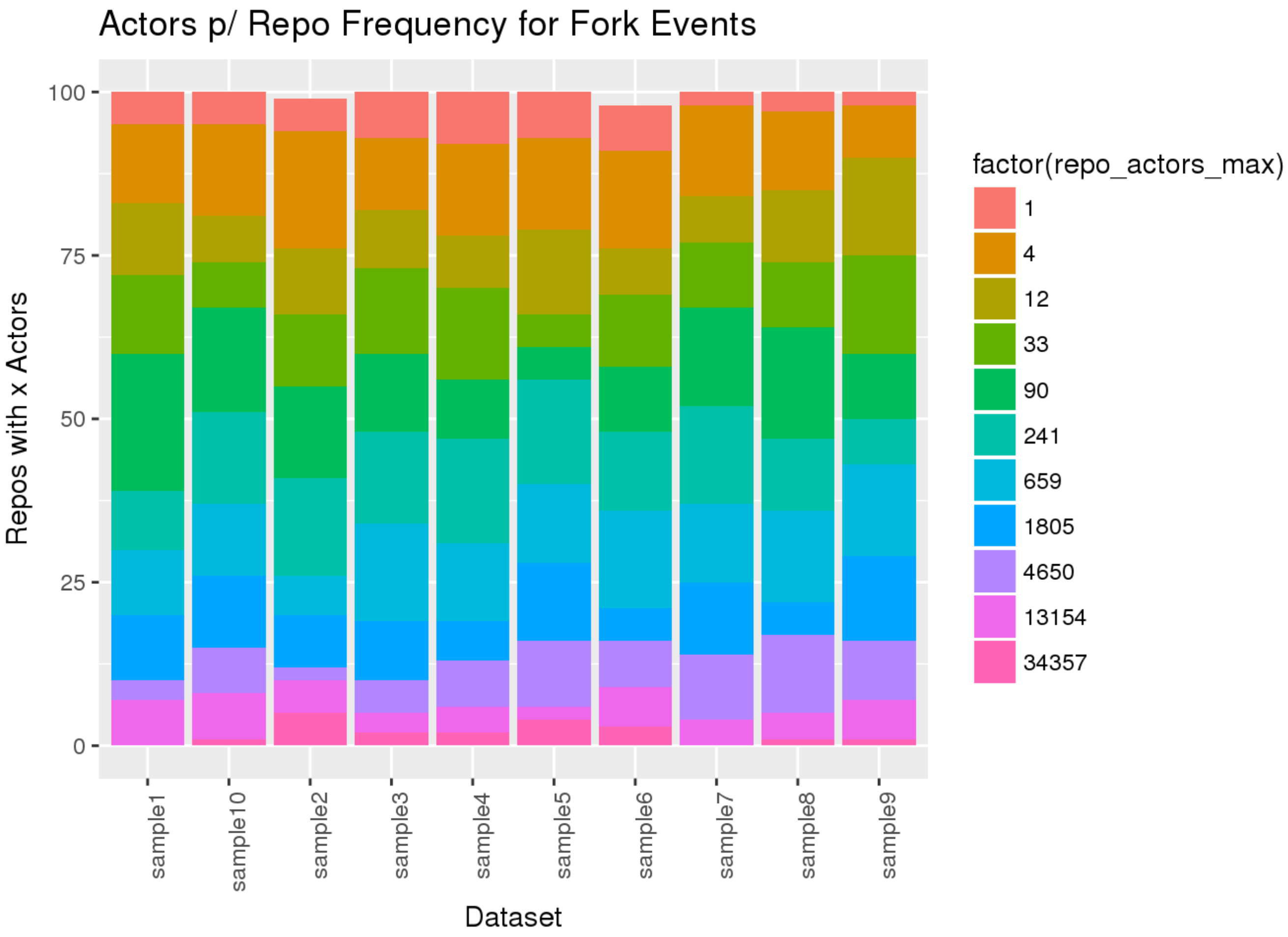


Actors p/ Repo Frequency for Watch Events

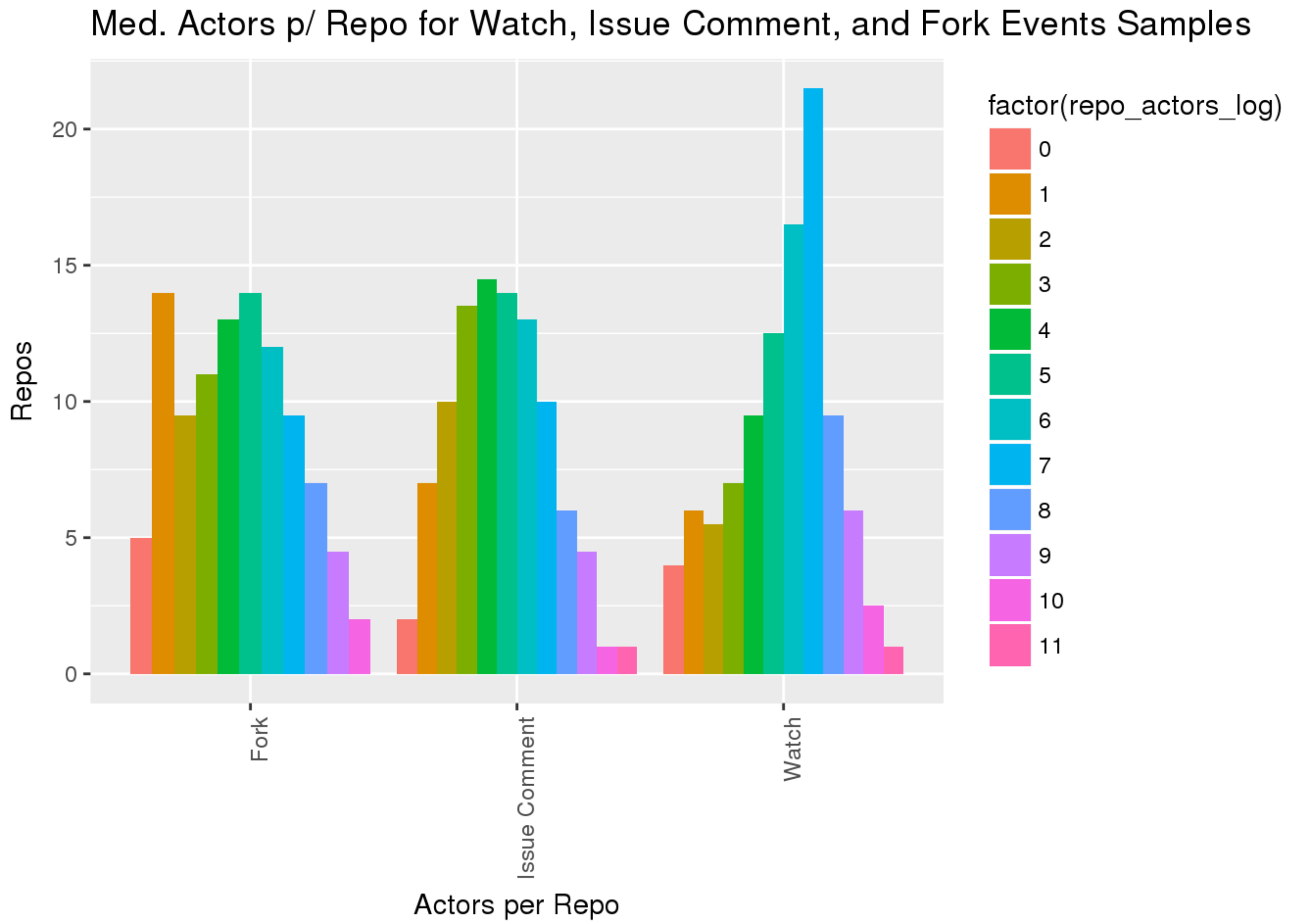
```r
isscomment_actors <- isscomment_repo_summary %>%
  group_by(dataset, repo_actors_log) %>%
  summarise(repo_count = n(),
            num_repo_actors_min = min(num_repo_actors),
            num_repo_actors_max = max(num_repo_actors)) %>%
  select(dataset, repo_actors_log, repo_count, num_repo_actors_max, num_repo_actors_mi
n)

isscomment_actors_log <- isscomment_actors %>%
  group_by(repo_actors_log) %>%
  summarise(repo_actors_max = max(num_repo_actors_max),
            repo_actors_min = min(num_repo_actors_min))

isscomment_actors <- merge(isscomment_actors, isscomment_actors_log, by="repo_actors_l
og")

isscomment_actors <- isscomment_actors[order(isscomment_actors$repo_count,
decreasing=TRUE),]

ggplot(data = isscomment_actors,
       aes(x = dataset,
           y = repo_count,
           fill=factor(repo_actors_max))) +
  geom_bar(stat="identity", position="stack") +
  ylab("Repos with x Actors") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Actors p/ Repo Frequency for Issue Comment Events")
```


Actors p/ Repo Frequency for Issue Comment Events
Repos with x Actors
100
75
50
25
0
sample1
sample10
sample2
sample3
sample4
sample5
sample6
sample7
sample8
sample9
Dataset
factor(repo_actors_max)
1
4
12
33
90
244
665
1795
4865
12940
28524
94620

```
fork_actors <- fork_repo_summary %>%
  group_by(dataset, repo_actors_log) %>%
  summarise(repo_count = n(),
            num_repo_actors_min = min(num_repo_actors),
            num_repo_actors_max = max(num_repo_actors)) %>%
  select(dataset, repo_actors_log, repo_count, num_repo_actors_max, num_repo_actors_mi
n)

fork_actors_log <- fork_actors %>%
  group_by(repo_actors_log) %>%
  summarise(repo_actors_max = max(num_repo_actors_max),
            repo_actors_min = min(num_repo_actors_min))

fork_actors <- merge(fork_actors, fork_actors_log, by="repo_actors_log")

fork_actors <- fork_actors[order(fork_actors$repo_count, decreasing=TRUE),]

ggplot(data = fork_actors,
       aes(x = dataset,
           y = repo_count,
           fill=factor(repo_actors_max))) +
  geom_bar(stat="identity", position="stack") +
  ylab("Repos with x Actors") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Actors p/ Repo Frequency for Fork Events")
```


Actors p/ Repo Frequency for Fork Events
Repos with x Actors
100
75
50
25
0
sample1
sample10
sample2
sample3
sample4
sample5
sample6
sample7
sample8
sample9
Dataset
factor(repo_actors_max)
1
4
12
33
90
241
659
1805
4650
13154
34357

```
watch_actors_med <- watch_actors %>%
  group_by(repo_actors_log) %>%
  summarise(dataset = "Watch",
    repo_count = median(repo_count),
    repo_actors_max = median(repo_actors_max))

isscomment_actors_med <- isscomment_actors %>%
  group_by(repo_actors_log) %>%
  summarise(dataset =  "Issue Comment",
    repo_count = median(repo_count),
    repo_actors_max = median(repo_actors_max))

fork_actors_med <- fork_actors %>%
  group_by(repo_actors_log) %>%
  summarise(dataset = "Fork",
    repo_count = median(repo_count),
    repo_actors_max = median(repo_actors_max))

high_type_samples_actors <- bind_rows(watch_actors_med, isscomment_actors_med, fork_ac
tors_med)

ggplot(data = high_type_samples_actors,
       aes(x=dataset, y=repo_count, fill=factor(repo_actors_log))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Actors per Repo") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Med. Actors p/ Repo for Watch, Issue Comment, and Fork Events Samples")
```



```
ggsave(filename="high_type_samples_actors.png")
```

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

# Medium Participation

Medium appears to be split between High and Low and should not be its own category. Release events might provide insight into how to potentially readjust our participation rate calculation to include a Medium category, of if one exists at all.

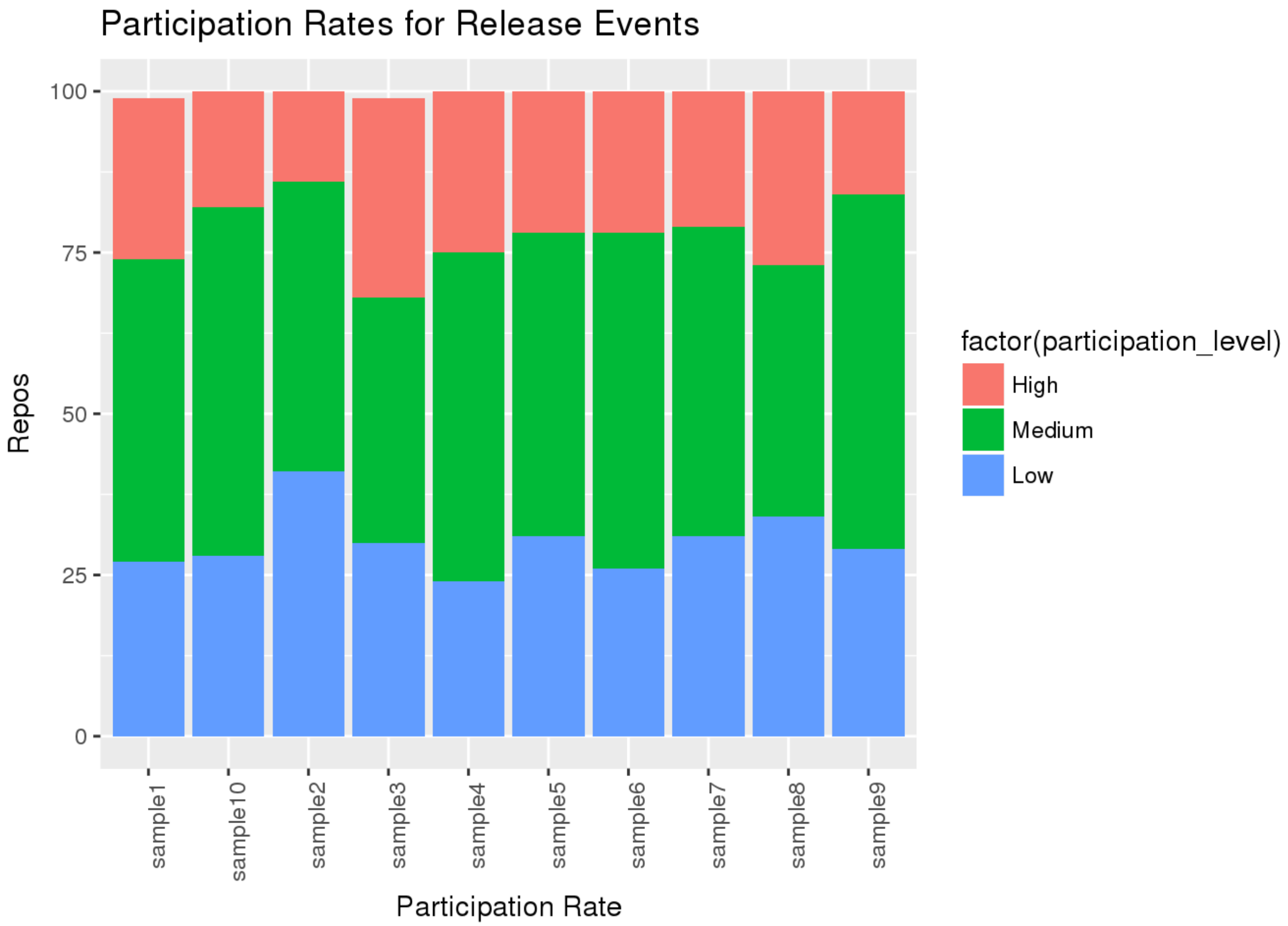
Release events occur frequently with Medium participation repositories therefore samples taken from these events would have a higher probability of containing this type of repository. Member events occur most frequently but they do not represent a significant enough proportion of event activity to warrant analysis.

```
release_events_repo_samples <- readRDS("release_events_repo_samples.rds")

release_events_repo_samples <-
    mutate(release_events_repo_samples,
           participation_level = ifelse(participation_rate < 1 & participation_rate >
0, 'Medium', ''))

release_events_repo_samples <-
  mutate(release_events_repo_samples, participation_level = ifelse(participation_rate
== 0, 'High', participation_level))

release_events_repo_samples <-
  mutate(release_events_repo_samples, participation_level = ifelse(participation_rate
== 1, 'Low', participation_level))
```

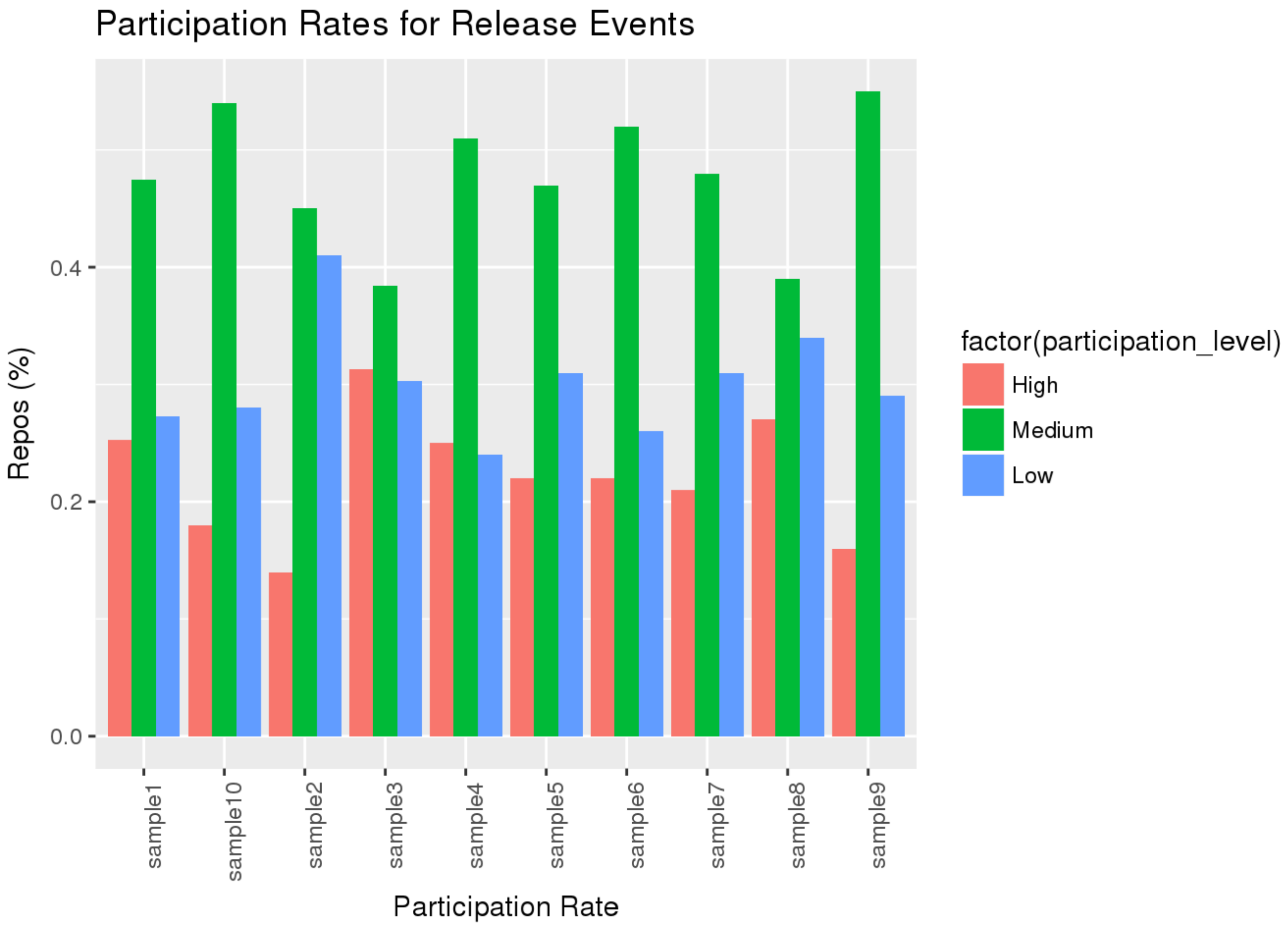
## Participation Rates

First we examine if the samples drawn by event type show a higher proportion of medium participation repositories.

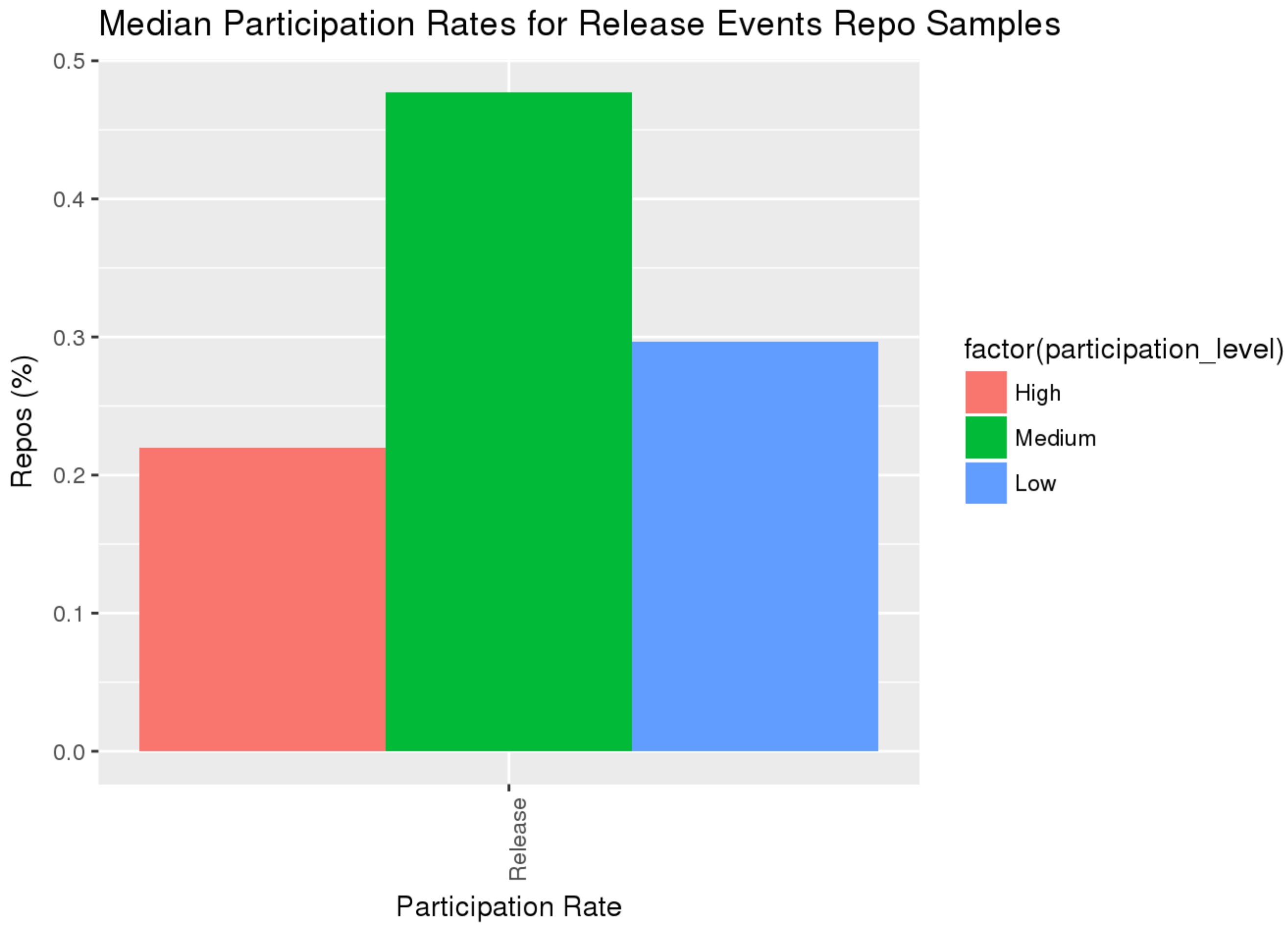
```
release_repo_summary <- release_events_repo_samples %>%
  group_by(dataset, repo_name) %>%
  summarise(
    participation_level = max(participation_level),
    num_repo_events = max(num_repo_events),
    num_repo_actors = max(num_repo_actors),
    repo_actors_log = round(log(num_repo_actors)),
    repo_events_log = round(log(num_repo_events))
  )

release_dataset_summary <- release_repo_summary %>%
  group_by(dataset) %>%
  summarise(repos_in_dataset = n(),
            actors_in_dataset = sum(num_repo_actors),
            events_in_dataset = sum(num_repo_events))

release_participation_summary <- release_repo_summary %>%
  group_by(dataset, participation_level) %>%
  summarise(num_repos = n())

release_participation_summary <- merge(release_participation_summary, release_dataset_
summary, by="dataset")

release_participation_summary <- release_participation_summary %>%
  mutate(repos_perc = num_repos/repos_in_dataset)

release_participation_summary$participation_level <- factor(release_participation_summ
ary$participation_level,
                           levels = c("High", "Medium", "Low"))

ggplot(data = release_participation_summary,
       aes(x=dataset, y=num_repos, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="stack") +
  xlab("Participation Rate") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Release Events")
```



```
ggplot(data = release_participation_summary,
       aes(x=dataset, y=repos_perc, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Release Events")
```



```
med_type_samples_pr <- release_participation_summary %>%
  group_by(participation_level) %>%
  summarise(num_repos_med = median(num_repos),
            num_repos_mean = mean(num_repos),
            repos_perc_med = median(repos_perc),
            repos_perc_mean = mean(repos_perc),
            dataset = "Release")

ggplot(data = med_type_samples_pr,
       aes(x=dataset, y=repos_perc_med, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Median Participation Rates for Release Events Repo Samples")
```



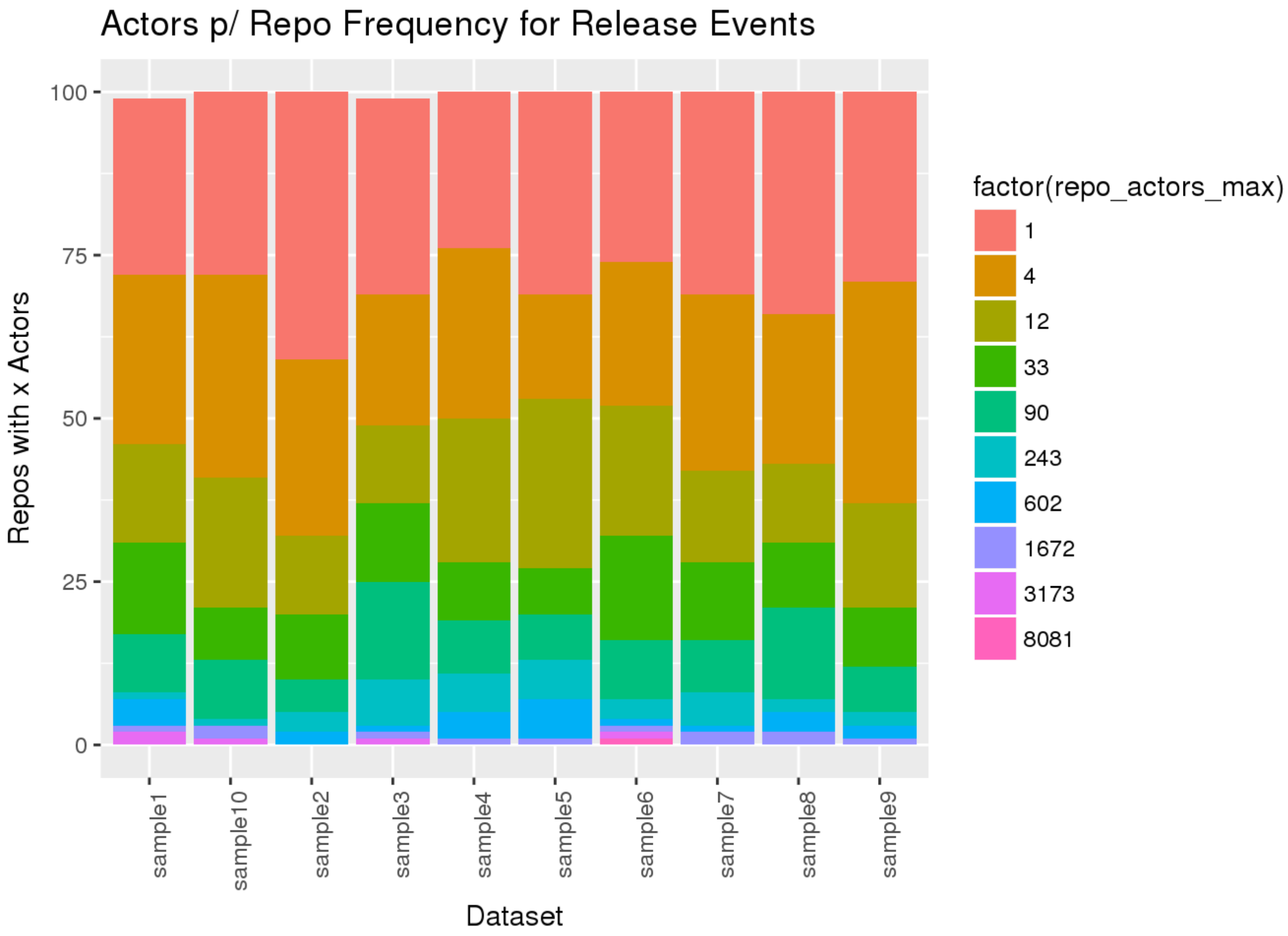
```
ggsave(filename="med_type_samples_participation.png")
```

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

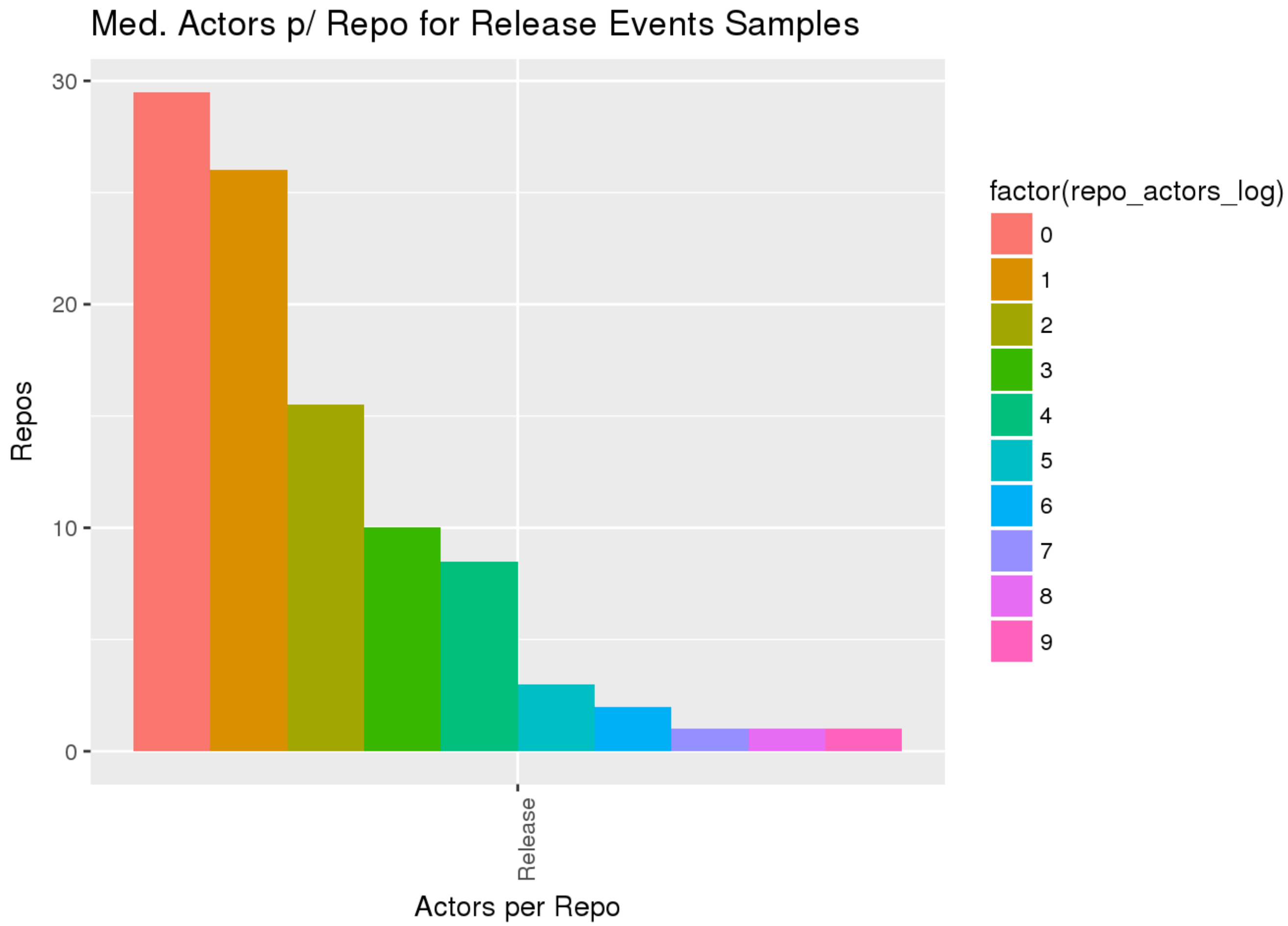
## Actors Per Repo

Second, we look at the number of unique actors that generated events and compare that to the medium participation population studied earlier.

```
release_actors <- release_repo_summary %>%
  group_by(dataset, repo_actors_log) %>%
  summarise(repo_count = n(),
            num_repo_actors_min = min(num_repo_actors),
            num_repo_actors_max = max(num_repo_actors)) %>%
  select(dataset, repo_actors_log, repo_count, num_repo_actors_max, num_repo_actors_mi
n)

release_actors_log <- release_actors %>%
  group_by(repo_actors_log) %>%
  summarise(repo_actors_max = max(num_repo_actors_max),
            repo_actors_min = min(num_repo_actors_min))

release_actors <- merge(release_actors, release_actors_log, by="repo_actors_log")

release_actors <- release_actors[order(release_actors$repo_count, decreasing=TRUE),]

ggplot(data = release_actors,
       aes(x = dataset,
           y = repo_count,
           fill=factor(repo_actors_max))) +
  geom_bar(stat="identity", position="stack") +
  ylab("Repos with x Actors") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Actors p/ Repo Frequency for Release Events")
```



```
med_type_samples_actors <- release_actors %>%
  group_by(repo_actors_log) %>%
  summarise(dataset = "Release",
    repo_count = median(repo_count),
    repo_actors_max = median(repo_actors_max))

ggplot(data = med_type_samples_actors,
       aes(x=dataset, y=repo_count, fill=factor(repo_actors_log))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Actors per Repo") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Med. Actors p/ Repo for Release Events Samples")
```



```
ggsave(filename="med_type_samples_actors.png")
```

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

# Low Participation

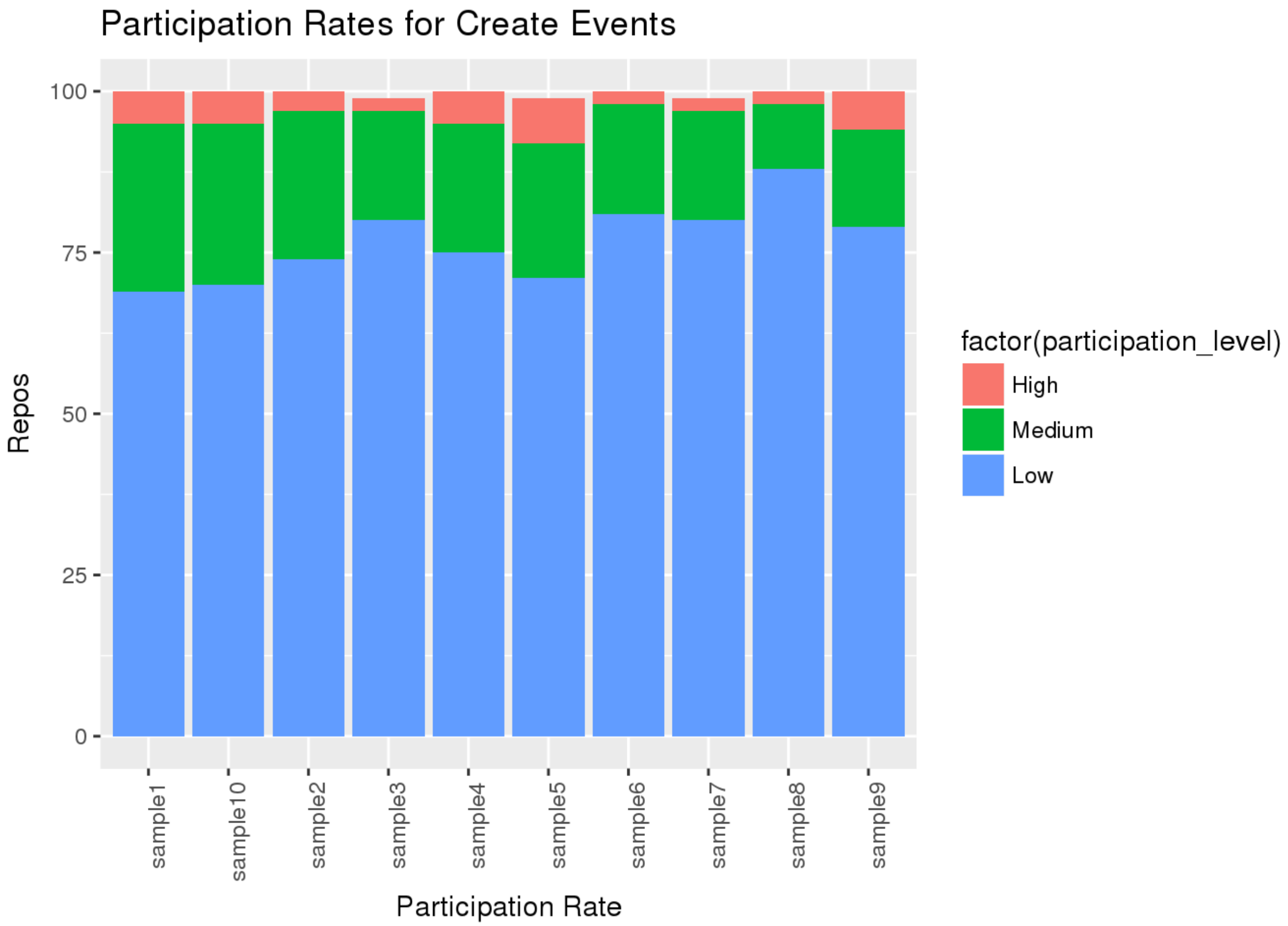
Create events show the highest association with Medium and Low repositories, therefore samples taken from these events would have a higher probability of containing these types of repositories.

```
create_events_repo_samples <- readRDS("create_events_repo_samples.rds")

create_events_repo_samples <-
    mutate(create_events_repo_samples,
           participation_level = ifelse(participation_rate < 1 & participation_rate >
0, 'Medium', ''))

create_events_repo_samples <-
  mutate(create_events_repo_samples, participation_level = ifelse(participation_rate =
= 0, 'High', participation_level))

create_events_repo_samples <-
  mutate(create_events_repo_samples, participation_level = ifelse(participation_rate =
= 1, 'Low', participation_level))
```

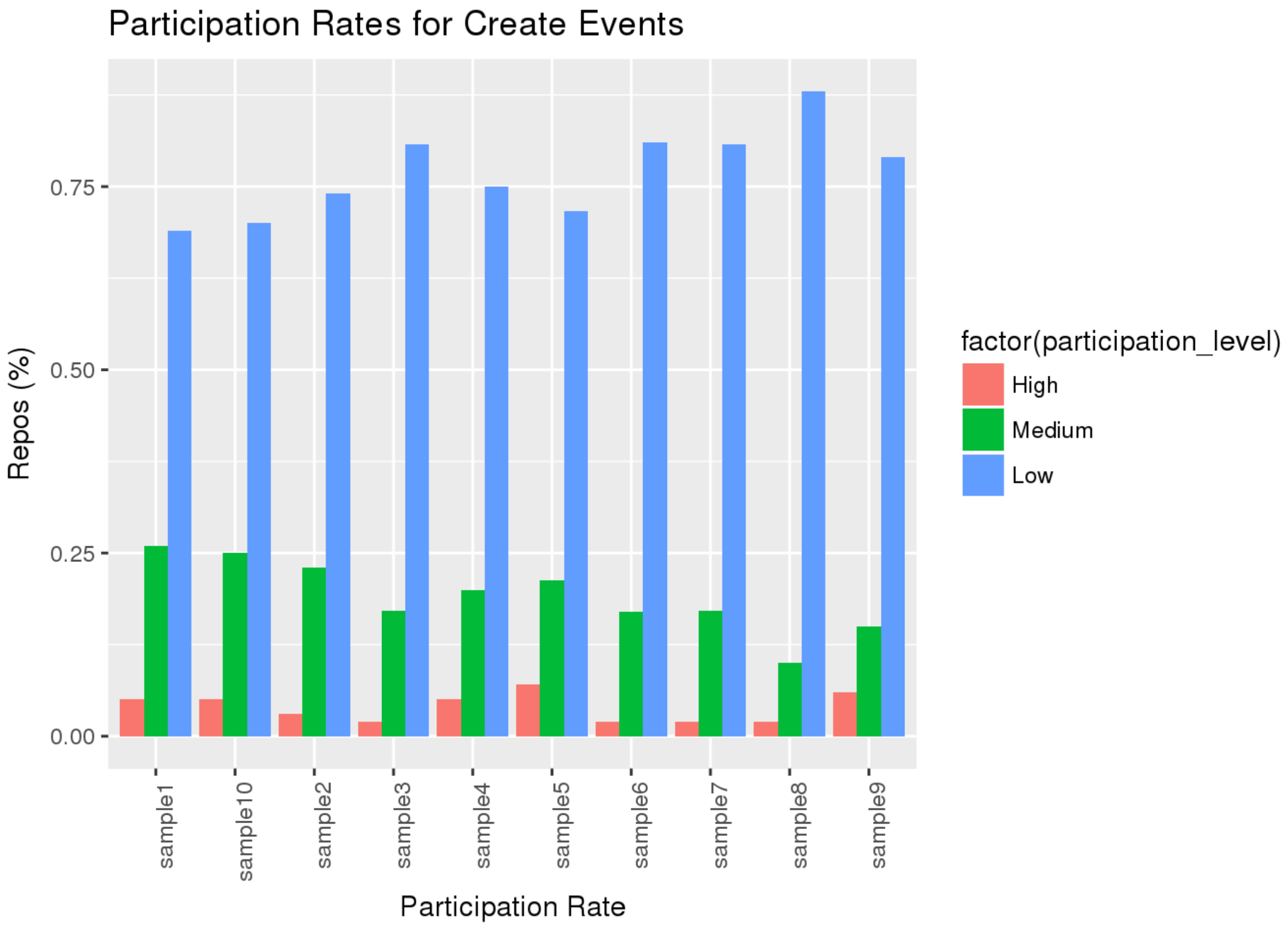
# Participation Rates

First we examine if the samples drawn by event type show a higher proportion of low participation repositories.

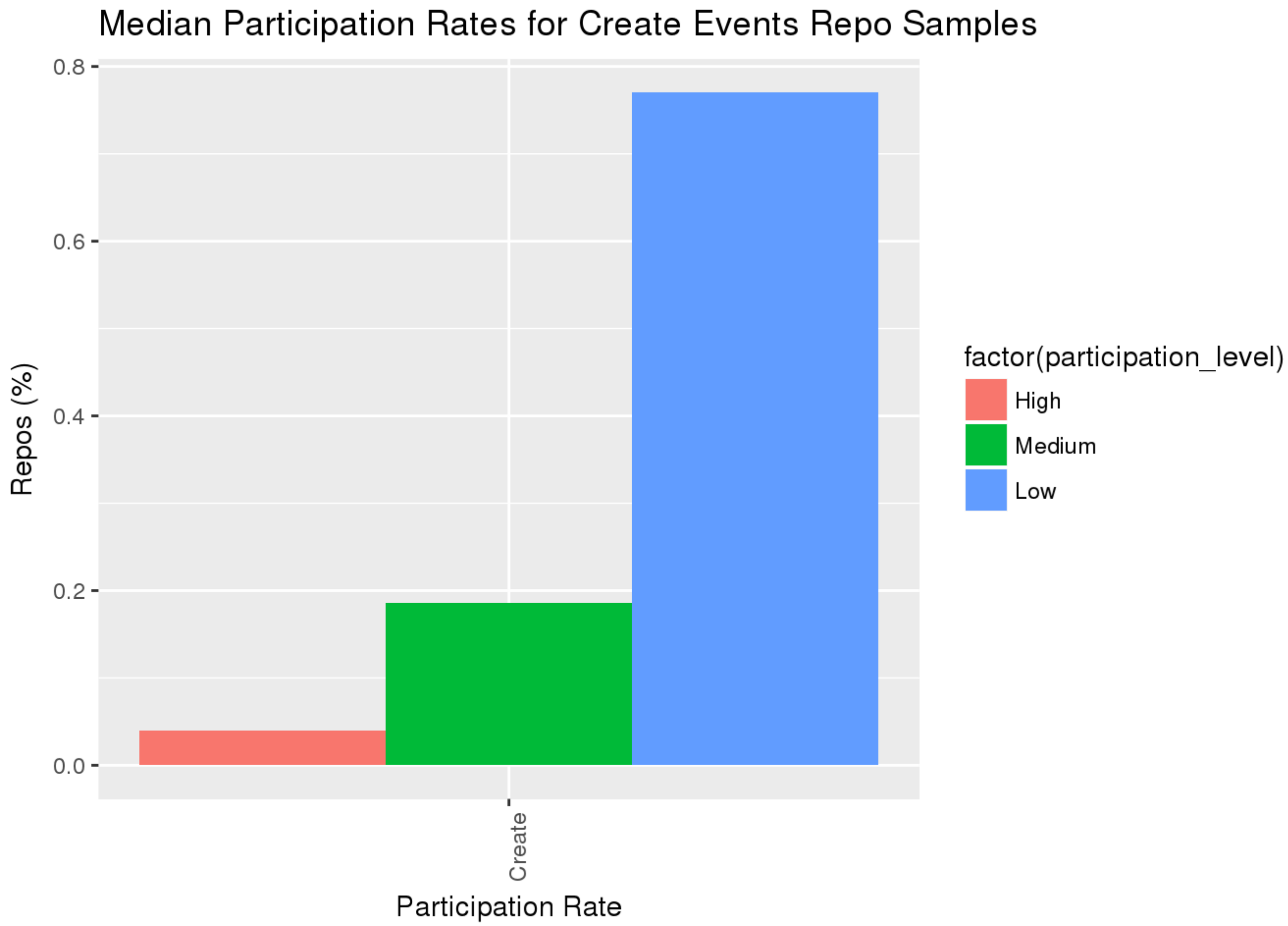
```
create_repo_summary <- create_events_repo_samples %>%
  group_by(dataset, repo_name) %>%
  summarise(
    participation_level = max(participation_level),
    num_repo_events = max(num_repo_events),
    num_repo_actors = max(num_repo_actors),
    repo_actors_log = round(log(num_repo_actors)),
    repo_events_log = round(log(num_repo_events))
  )

create_dataset_summary <- create_repo_summary %>%
  group_by(dataset) %>%
  summarise(repos_in_dataset = n(),
            actors_in_dataset = sum(num_repo_actors),
            events_in_dataset = sum(num_repo_events))

create_participation_summary <- create_repo_summary %>%
  group_by(dataset, participation_level) %>%
  summarise(num_repos = n())

create_participation_summary <- merge(create_participation_summary, create_dataset_sum
mary, by="dataset")

create_participation_summary <- create_participation_summary %>%
  mutate(repos_perc = num_repos/repos_in_dataset)

create_participation_summary$participation_level <- factor(create_participation_summar
y$participation_level,
                           levels = c("High", "Medium", "Low"))

ggplot(data = create_participation_summary,
       aes(x=dataset, y=num_repos, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="stack") +
  xlab("Participation Rate") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Create Events")
```



```
ggplot(data = create_participation_summary,
       aes(x=dataset, y=repos_perc, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Participation Rates for Create Events")
```



```
low_type_samples_pr <- create_participation_summary %>%
  group_by(participation_level) %>%
  summarise(num_repos_med = median(num_repos),
            num_repos_mean = mean(num_repos),
            repos_perc_med = median(repos_perc),
            repos_perc_mean = mean(repos_perc),
            dataset = "Create")

ggplot(data = low_type_samples_pr,
       aes(x=dataset, y=repos_perc_med, fill=factor(participation_level))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Participation Rate") +
  ylab("Repos (%)")  +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Median Participation Rates for Create Events Repo Samples")
```



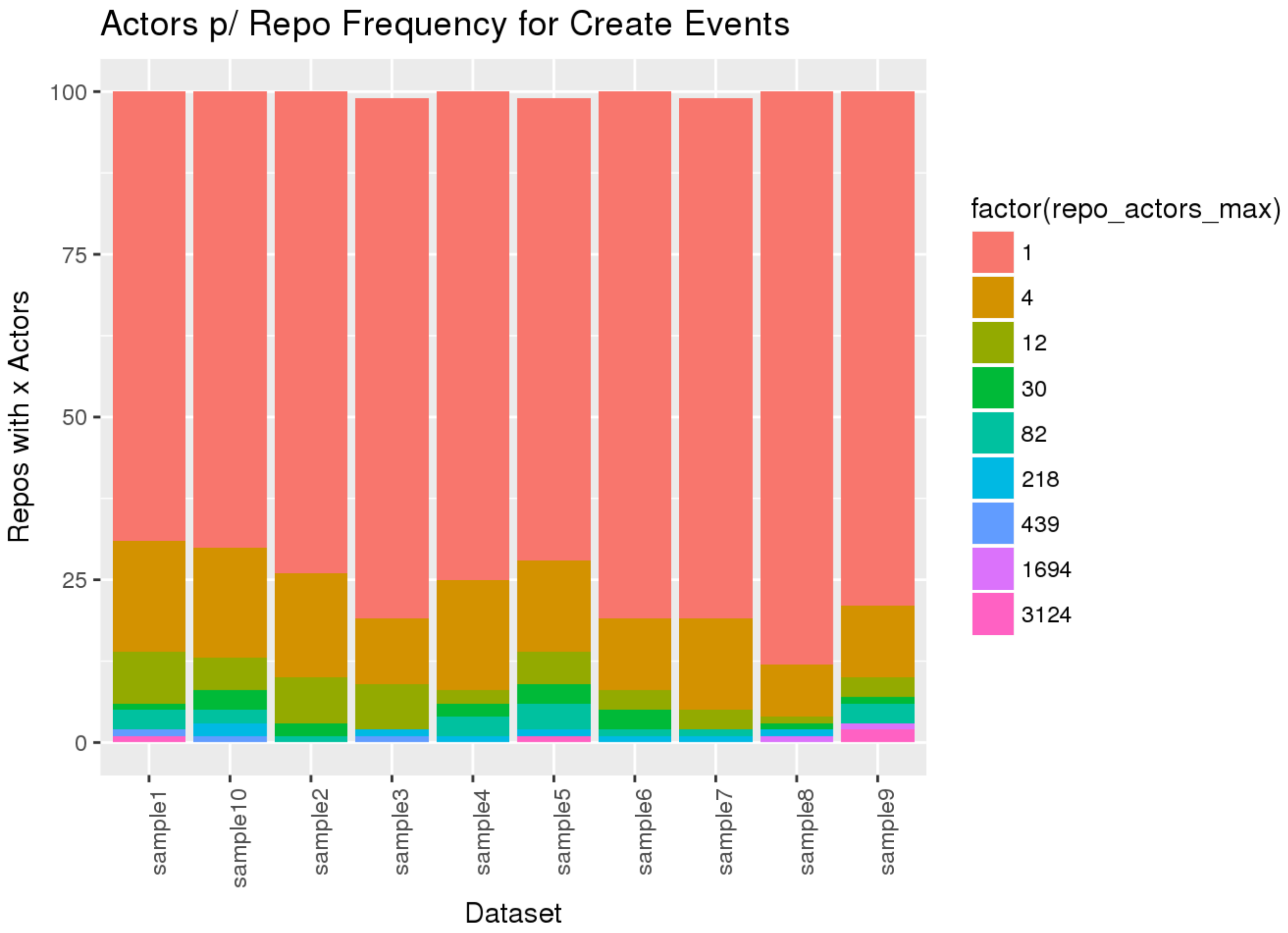
```
ggsave(filename="low_type_samples_participation.png")
```

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

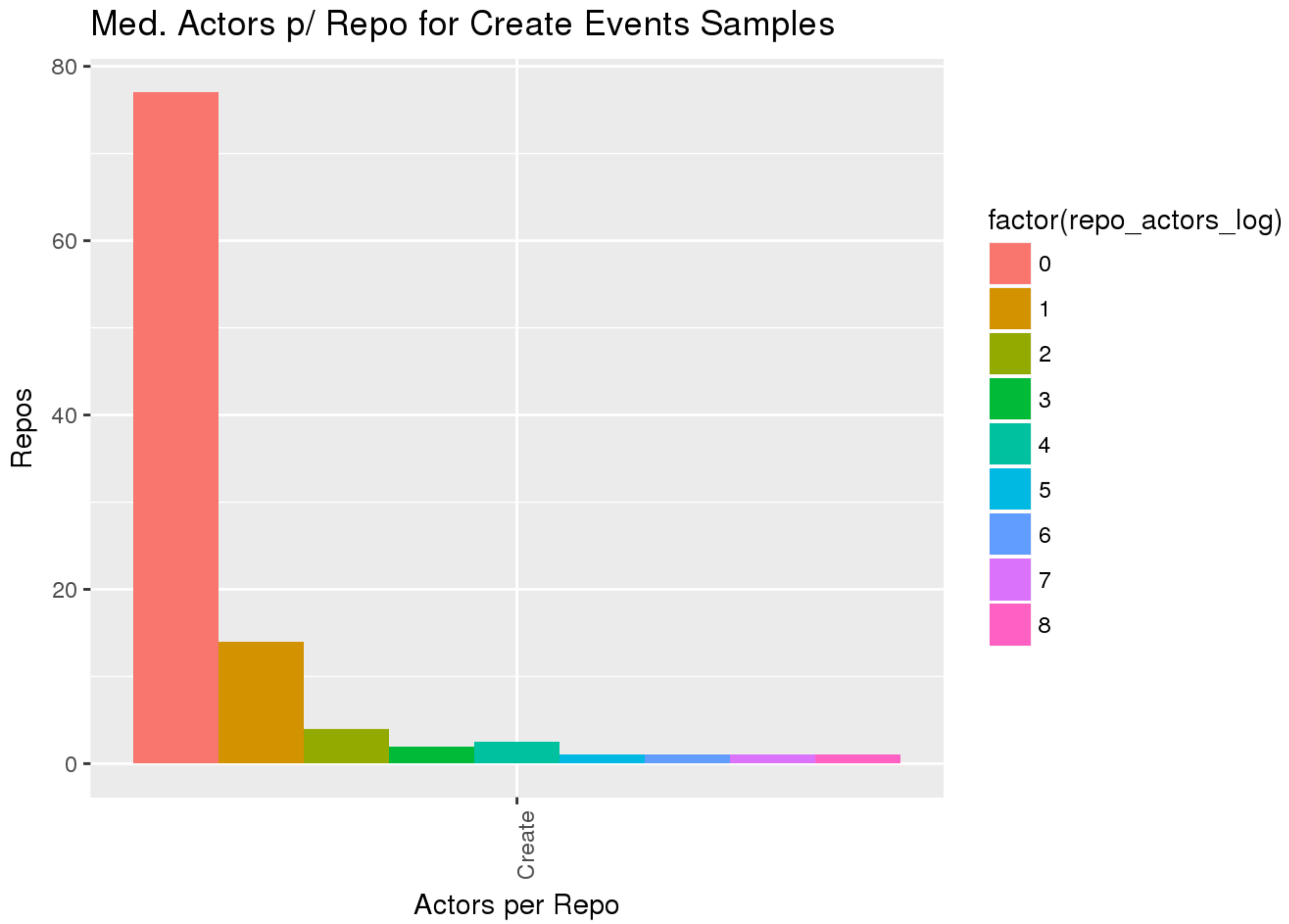
## Actors Per Repo

Second, we look at the number of unique actors that generated events and compare that to the low participation population studied earlier.

```
create_actors <- create_repo_summary %>%
  group_by(dataset, repo_actors_log) %>%
  summarise(repo_count = n(),
            num_repo_actors_min = min(num_repo_actors),
            num_repo_actors_max = max(num_repo_actors)) %>%
  select(dataset, repo_actors_log, repo_count, num_repo_actors_max, num_repo_actors_mi
n)

create_actors_log <- create_actors %>%
  group_by(repo_actors_log) %>%
  summarise(repo_actors_max = max(num_repo_actors_max),
            repo_actors_min = min(num_repo_actors_min))

create_actors <- merge(create_actors, create_actors_log, by="repo_actors_log")

create_actors <- create_actors[order(create_actors$repo_count, decreasing=TRUE),]

ggplot(data = create_actors,
       aes(x = dataset,
           y = repo_count,
           fill=factor(repo_actors_max))) +
  geom_bar(stat="identity", position="stack") +
  ylab("Repos with x Actors") +
  xlab("Dataset") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Actors p/ Repo Frequency for Create Events")
```



```
low_type_samples_actors <- create_actors %>%
  group_by(repo_actors_log) %>%
  summarise(dataset = "Create",
    repo_count = median(repo_count),
    repo_actors_max = median(repo_actors_max))

ggplot(data = low_type_samples_actors,
       aes(x=dataset, y=repo_count, fill=factor(repo_actors_log))) +
  geom_bar(stat="identity", position="dodge") +
  xlab("Actors per Repo") +
  ylab("Repos") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Med. Actors p/ Repo for Create Events Samples")
```



```
ggsave(filename="low_type_samples_actors.png")
```

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

# Conclusions

The data used in this study is insufficient to conclude if participation rate (the proportion of events per actor based on events data) provides an accurate metric for categorizing GitHub repositories. However we have clearly proven that some event types have a higher frequency of repositories with certain levels of participation. Therefore, random samples based on an event type most closely correlated with a given participation level will result in a higher proportion of repositories that are representative or very close to that participation level.

The data used in this study is insufficient to conclude if stratifying GitHub repositories by participation rate reduces variability. While the analysis in this study indicates that it does reduce variability somewhat for the parameters studied, these parameters were used to make the participation rate calculation in the first place. Therefore, further analysis is needed on parameters that are independent of the participation rate calculation.

One issue that came up in the study was the participation rate calculation itself. It is not clear how accurate this calculation is. The intent of this study was to propose the simplest model by which to stratify the repository to look for common themes. This calculation does not take into consideration activity over time and it treats all events as having equal participation value. Further analysis is needed to actually look at the data available on Github for repositories that fall into these participation levels. The main impacts this would have on this study is the potential for skewing the early results.

Finally, it's not clear whether "Medium" participation represents its own category or if it is actually a subtype. Further analysis on the Github repositories themselves may provide more insight into this. Also refactoring the participation calculation itself may provide a more accurate "Medium" range.

Depsite these concerns, we can conclude with confidence that the event type correlation is accurate because we evaluated the number of unique event actors per repo and saw a definite shift in frequency distribution. For a preliminary analysis intent on determining a methodology for sampling, the analysis is sufficiently conclusive to move on to the next phase.

The next step in this process is to test our hypotheses by looking at actual GitHub repository data to see how it compares. Because reconstructing this data to match our time frame would introduce complexity, the next study will instead pull recent samples based on the event types explored in this study.