Introduction

This notebook explores an A/B test conducted in the mobile puzzle game Cookie Cats. The game developers experimented with moving a progression gate from level 30 to level 40 in order to understand whether this change affects short-term and long-term player retention. The goal of this analysis is to compare user behavior across the two versions of the game and form a data-driven conclusion about whether the new gate placement improves retention or unintentionally harms engagement. The dataset contains player activity logs, retention outcomes, and the assigned experimental group. Through exploratory analysis, we aim to understand the structure of the data, compare retention patterns across versions, and prepare the ground for statistical testing.

```
options(warn = -1)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   4.0.0     ✓ tibble    3.3.0
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.1.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(dplyr)
library(broom)
library(infer)
library(ggplot2)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
df <- read_csv("cookie_cats.csv") %>% clean_names()
```

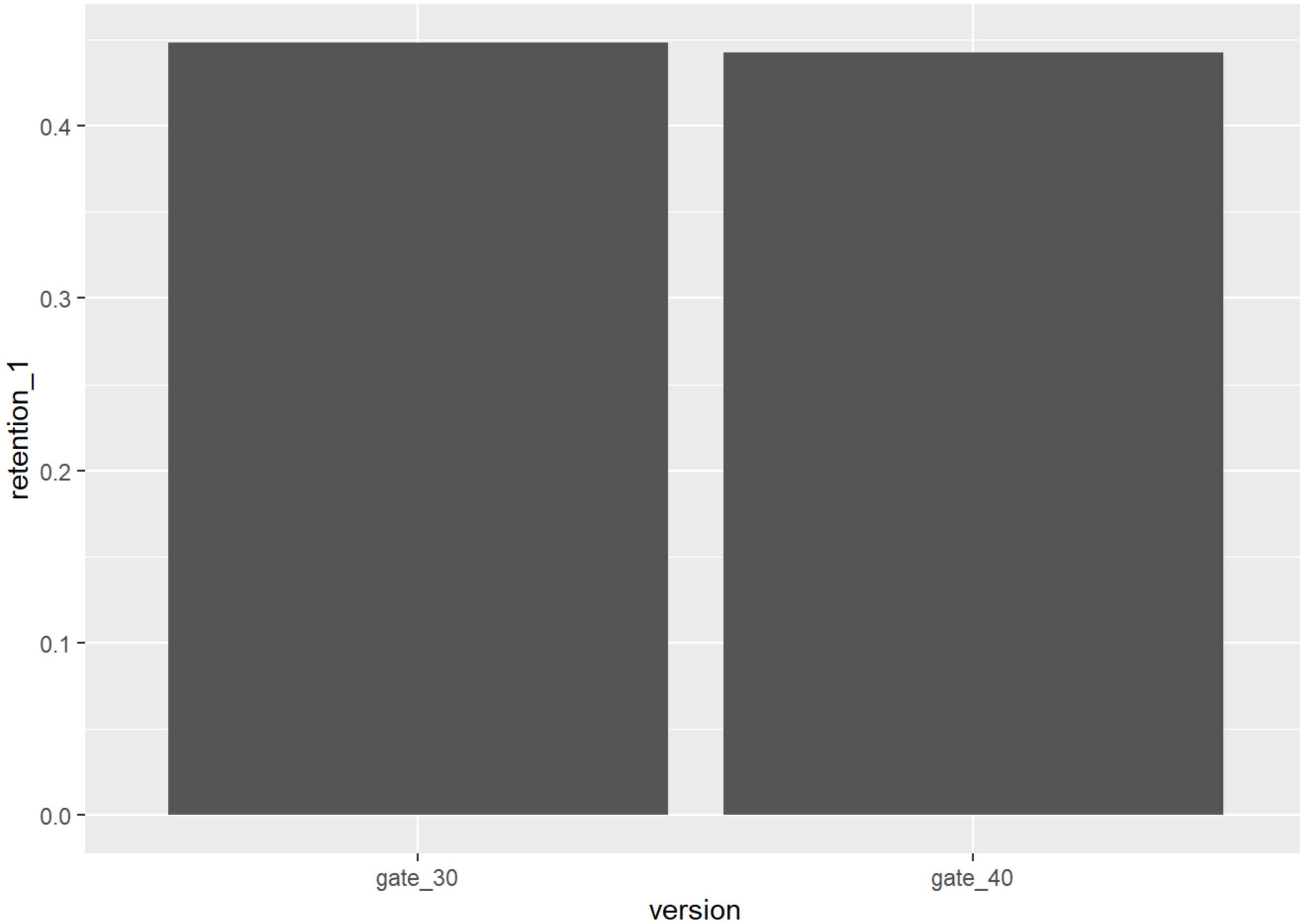
```
## Rows: 90189 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): version
## dbl (2): userid, sum_gamerounds
## lgl (2): retention_1, retention_7
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df$version <- factor(df$version, levels=c("gate_30","gate_40"))
```

Exploratory Data Analysis

Understanding retention differences at a high level. This initial plot compares the average one-day retention rates between the two game versions to give an early indication of how the gate change may influence engagement.

```
df %>%
  group_by(version) %>%
  summarise(retention_1 = mean(retention_1),
            retention_7 = mean(retention_7)) %>%
  ggplot(aes(version, retention_1)) +
  geom_col()
```



This comparison shows whether the treatment group appears to retain better or worse on the first day, helping identify whether the difference looks meaningful enough for further investigation.

Inspecting the structure of the dataset. Understanding the variables and ensuring the dataset is clean is necessary before deeper analysis.

```
str(df)
```

```
## spc_tbl_ [90,189 × 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ userid        : num [1:90189] 116 337 377 483 488 ...
##  $ version       : Factor w/ 2 levels "gate_30","gate_40": 1 1 2 2 2 2 1 2 2 2 ...
##  $ sum_gamerounds: num [1:90189] 3 38 165 1 179 187 0 2 108 153 ...
##  $ retention_1   : logi [1:90189] FALSE TRUE TRUE FALSE TRUE TRUE ...
##  $ retention_7   : logi [1:90189] FALSE FALSE FALSE FALSE TRUE TRUE ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   userid = col_double(),
##   ..   version = col_character(),
##   ..   sum_gamerounds = col_double(),
##   ..   retention_1 = col_logical(),
##   ..   retention_7 = col_logical()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
head(df)
```

```
## # A tibble: 6 × 5
##   userid version sum_gamerounds retention_1 retention_7
##    <dbl> <fct>            <dbl> <lgl>       <lgl>
## 1    116 gate_30              3 FALSE       FALSE
## 2    337 gate_30             38 TRUE        FALSE
## 3    377 gate_40            165 TRUE        FALSE
## 4    483 gate_40              1 FALSE       FALSE
## 5    488 gate_40            179 TRUE        TRUE
## 6    540 gate_40            187 TRUE        TRUE
```

```r
summary(df)
```

```
##      userid             version      sum_gamerounds     retention_1
##  Min.   :    116   gate_30:44700   Min.   :    0.00   Mode :logical
##  1st Qu.:2512230   gate_40:45489   1st Qu.:    5.00   FALSE:50036
##  Median :4995815                   Median :   16.00   TRUE :40153
##  Mean   :4998412                   Mean   :   51.87
##  3rd Qu.:7496452                   3rd Qu.:   51.00
##  Max.   :9999861                   Max.   :49854.00
##  retention_7
##  Mode :logical
##  FALSE:73408
##  TRUE :16781
##
##
##
```

```r
colSums(is.na(df))
```

```
##         userid        version sum_gamerounds    retention_1    retention_7
##              0              0              0              0              0
```

```r
sapply(df, function(x) length(unique(x)))
```

```
##         userid        version sum_gamerounds    retention_1    retention_7
##          90189              2            942              2              2
```

The dataset appears clean, with no missing entries in key fields. The retention variables are binary, and the version variable contains two balanced categories, indicating readiness for A/B testing.

Checking group sizes. Balanced group sizes ensure that differences in outcomes are not driven by uneven exposure.

```r
df %>%
  count(version) %>%
  mutate(prop = n / sum(n))
```

```
## # A tibble: 2 × 3
##   version     n  prop
##   <fct>   <int> <dbl>
## 1 gate_30 44700 0.496
## 2 gate_40 45489 0.504
```

Group counts are nearly equal, confirming proper random assignment.

Summary of retention by group. This helps quantify central tendencies in retention behavior.

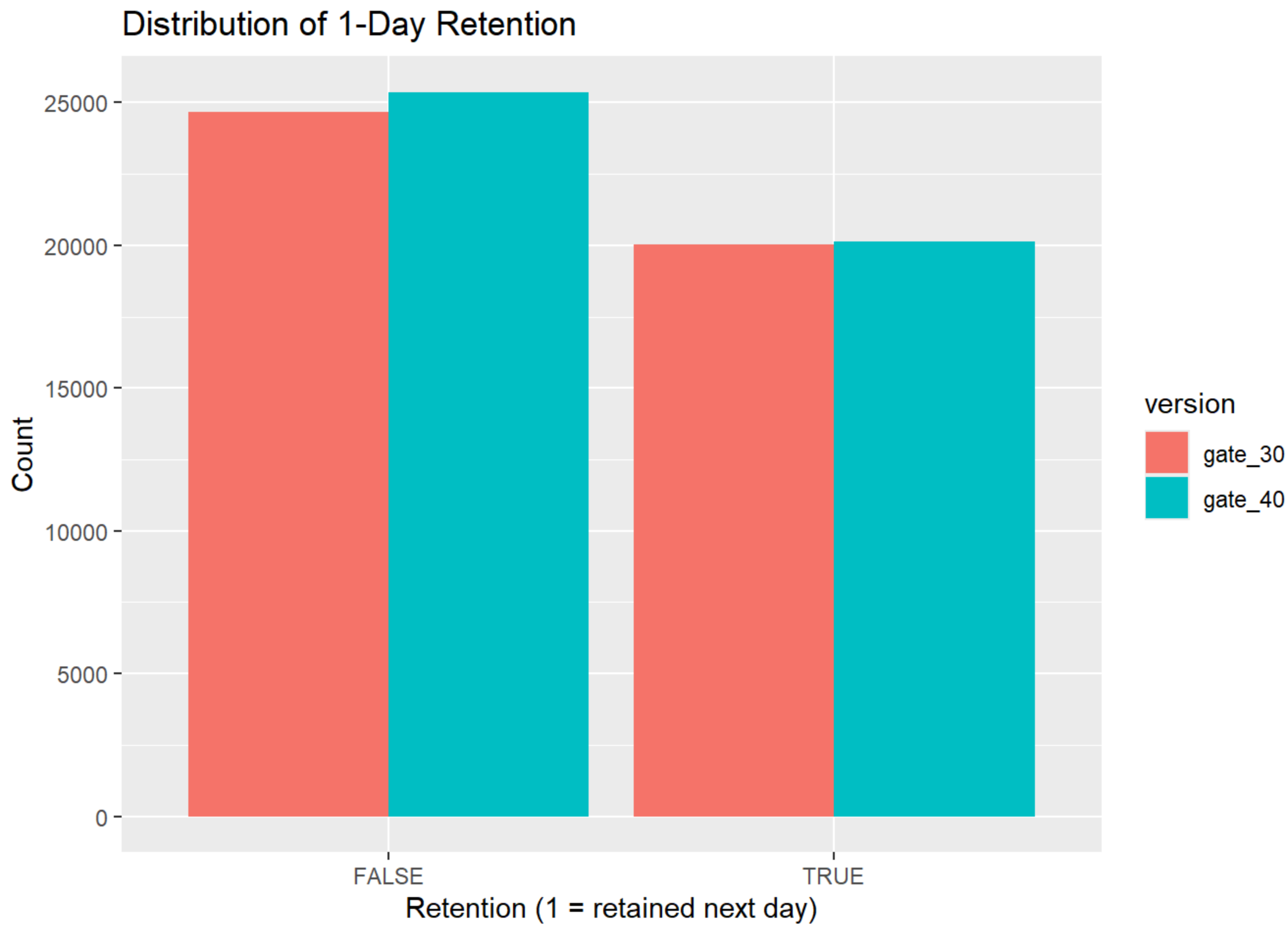
```r
df %>%
  group_by(version) %>%
  summarise(
    mean_retention_1 = mean(retention_1),
    mean_retention_7 = mean(retention_7),
    sd_retention_1 = sd(retention_1),
    sd_retention_7 = sd(retention_7),
    n = n()
  )
```

```
## # A tibble: 2 × 6
##   version mean_retention_1 mean_retention_7 sd_retention_1 sd_retention_7     n
##   <fct>              <dbl>            <dbl>          <dbl>          <dbl> <int>
## 1 gate_30            0.448            0.190          0.497          0.392 44700
## 2 gate_40            0.442            0.182          0.497          0.386 45489
```

The differences in means provide early clues about performance differences, while similar standard deviations confirm consistent variability across groups.

Distribution of 1-day retention. This visual highlights how many players returned on the next day.

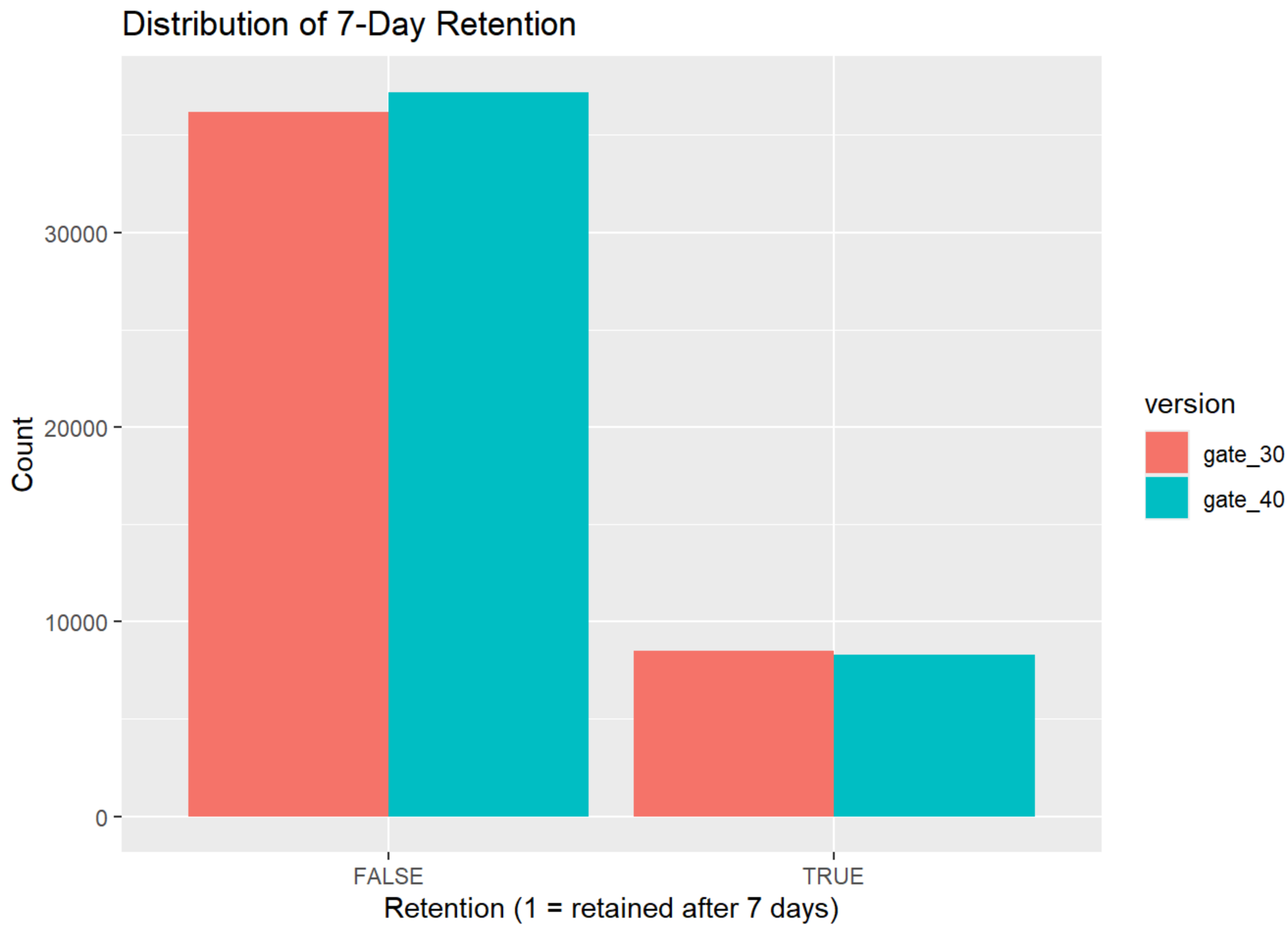
```r
ggplot(df, aes(x = retention_1, fill = version)) +
  geom_bar(position = "dodge") +
  labs(title = "Distribution of 1-Day Retention",
       x = "Retention (1 = retained next day)", y = "Count")
```



The bar heights reveal the relative proportions of retained and non-retained users across the two groups, giving an immediate impression of performance differences.

Distribution of 7-day retention. This plot examines longer-term engagement.

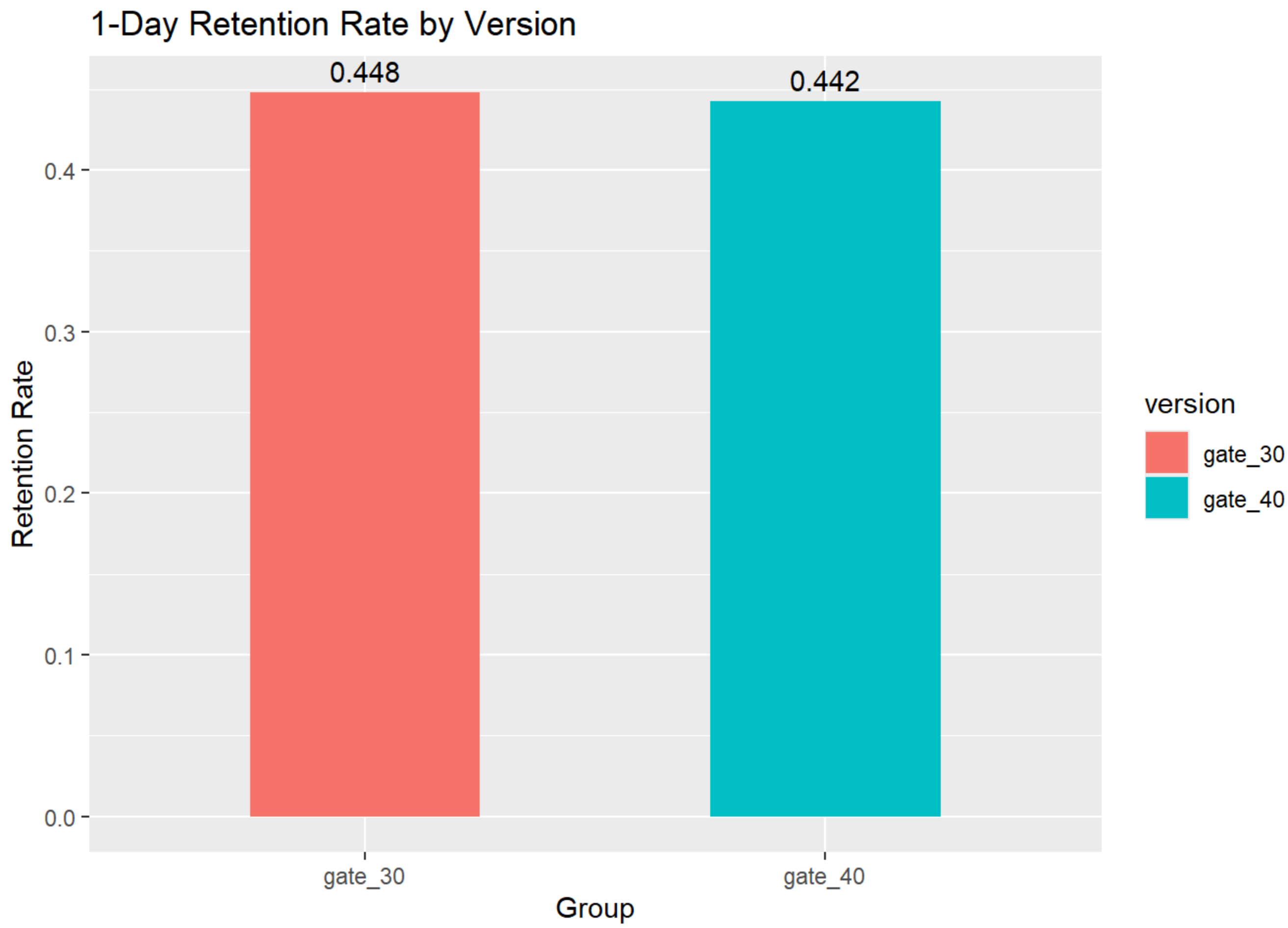
```
ggplot(df, aes(x = retention_7, fill = version)) +
  geom_bar(position = "dodge") +
  labs(title = "Distribution of 7-Day Retention",
       x = "Retention (1 = retained after 7 days)", y = "Count")
```



Seven-day retention is typically much lower, and visible differences between versions suggest whether the gate change affected deeper engagement patterns.

Comparing 1-day retention rates with numeric labels for clarity.

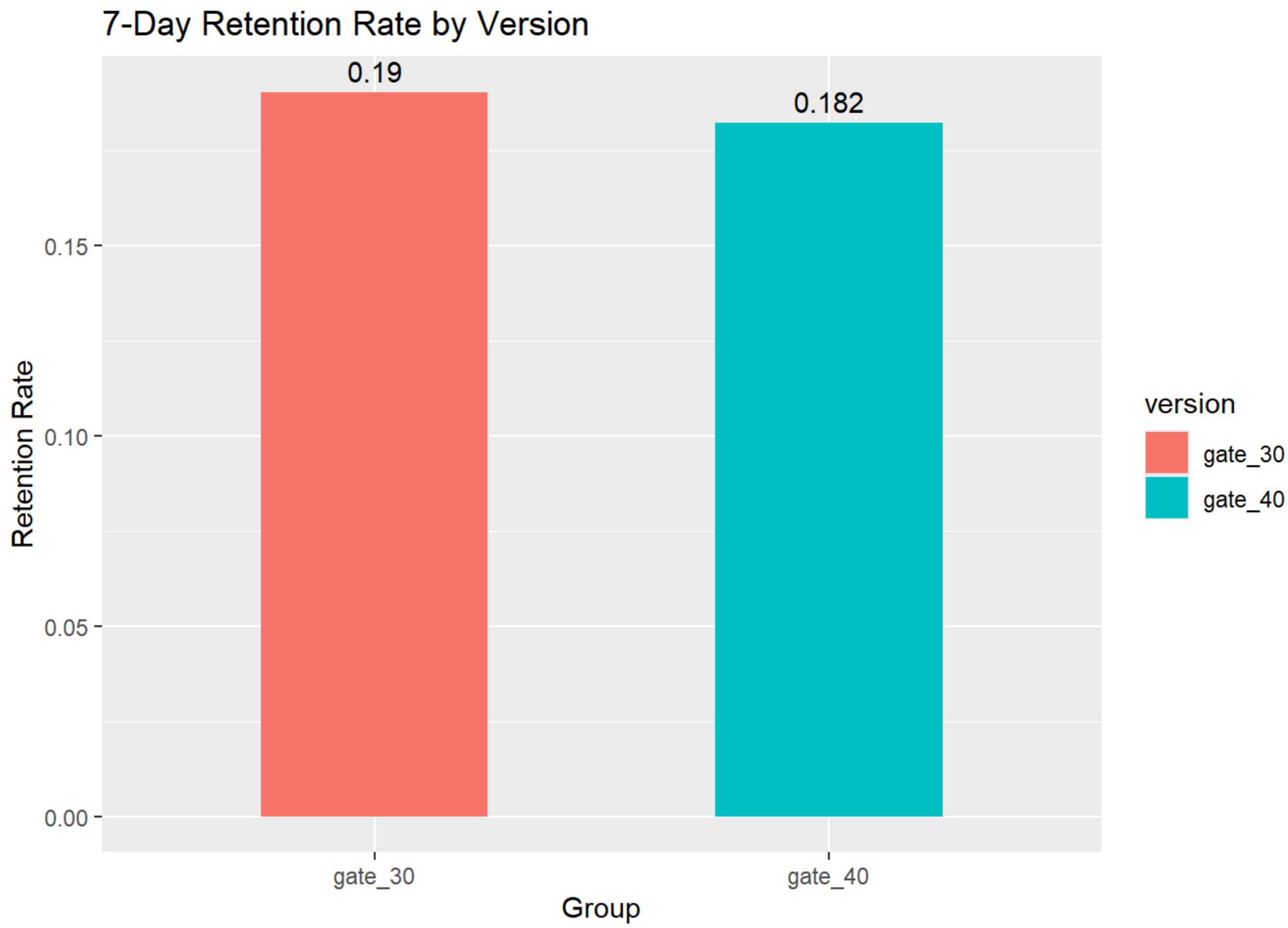
```
df %>%
  group_by(version) %>%
  summarise(rate = mean(retention_1)) %>%
  ggplot(aes(version, rate, fill = version)) +
  geom_col(width = 0.5) +
  geom_text(aes(label = round(rate, 3)), vjust = -0.5) +
  labs(title = "1-Day Retention Rate by Version",
       x = "Group", y = "Retention Rate")
```



The numeric labels make differences easier to interpret and help determine whether they justify formal testing.

Comparing 7-day retention rates in the same format for consistency.

```
df %>%
  group_by(version) %>%
  summarise(rate = mean(retention_7)) %>%
  ggplot(aes(version, rate, fill = version)) +
  geom_col(width = 0.5) +
  geom_text(aes(label = round(rate, 3)), vjust = -0.5) +
  labs(title = "7-Day Retention Rate by Version",
       x = "Group", y = "Retention Rate")
```



This metric reflects long-term engagement, offering deeper insight into the impact of the gate change.

Comparing both retention metrics side by side.

```
df_long <- df %>%
  pivot_longer(cols = c(retention_1, retention_7),
               names_to = "metric",
               values_to = "value")

ggplot(df_long, aes(x = version, y = value, fill = version)) +
  geom_bar(stat="summary", fun="mean", position="dodge") +
  facet_wrap(~ metric) +
  labs(title = "Comparison of Retention Metrics",
       y = "Mean Retention Rate")
```

Seeing both short- and long-term retention together helps distinguish whether the gate change affected early engagement, long-term engagement, or both.

User-level jitter plot to examine individual variation.

```
ggplot(df, aes(x = version, y = retention_1, color = version)) +
  geom_jitter(alpha = 0.2, width = 0.2) +
  labs(title = "User-Level Scatter of 1-Day Retention",
       y = "Retention")
```

The jitter spread shows variation in individual retention outcomes and confirms that no unusual clustering is influencing the results.

Correlation between 1-day and 7-day retention. This examines whether early engagement predicts long-term engagement.

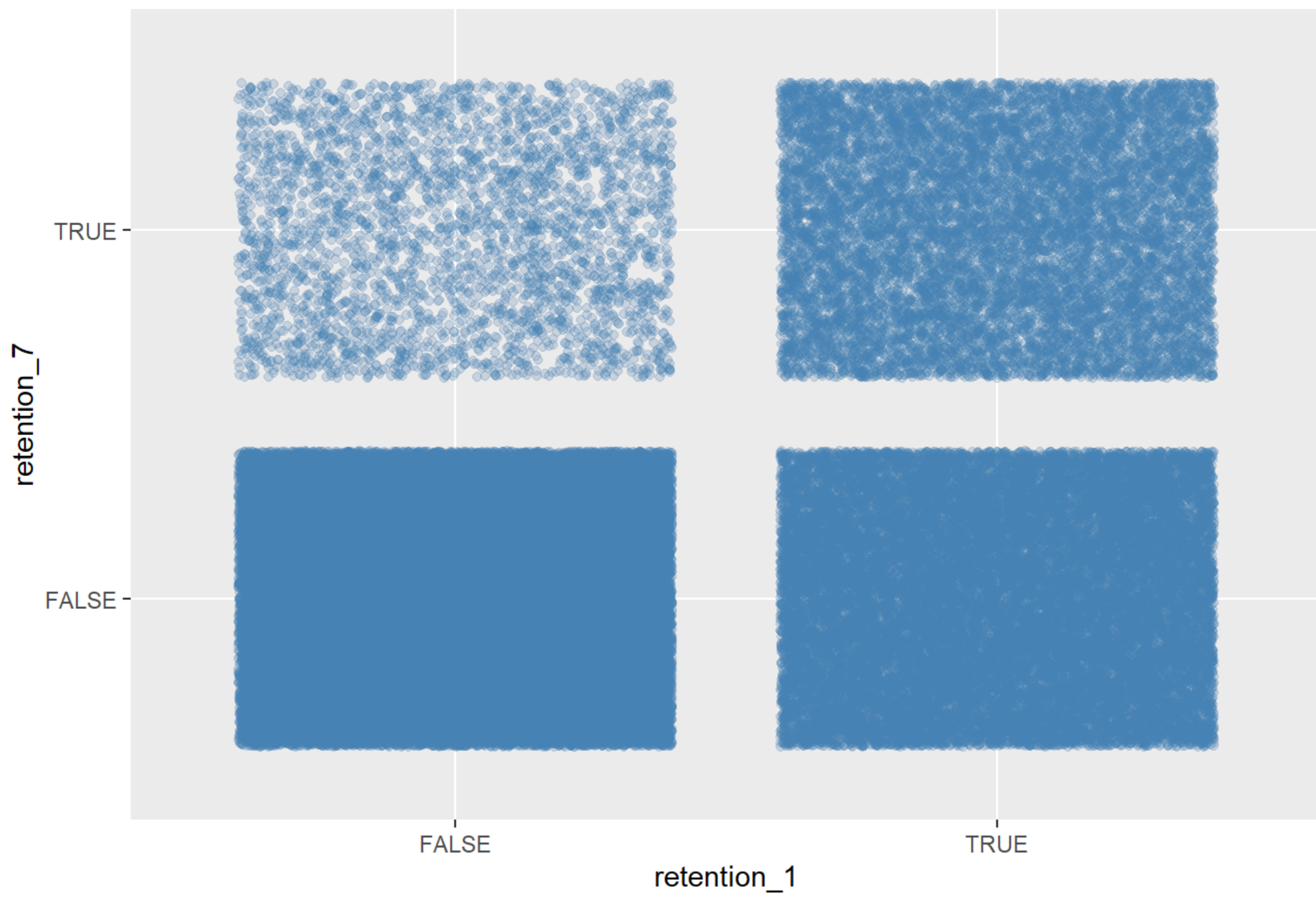
```
cor(df$retention_1, df$retention_7)
```

```
## [1] 0.3274012
```

```
ggplot(df, aes(x = retention_1, y = retention_7)) +
  geom_jitter(alpha = 0.2, color = "steelblue") +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Correlation Between Day-1 and Day-7 Retention")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Correlation Between Day-1 and Day-7 Retention



There is a clear relationship between returning on the first day and returning after a week, showing that early retention plays a strong role in predicting deeper engagement.