Analysis of Random Orderings for Brute Force Alg for One Sudoku Puzzle

Steven Kim

2025-11-21

Check for packages:

Read Data:

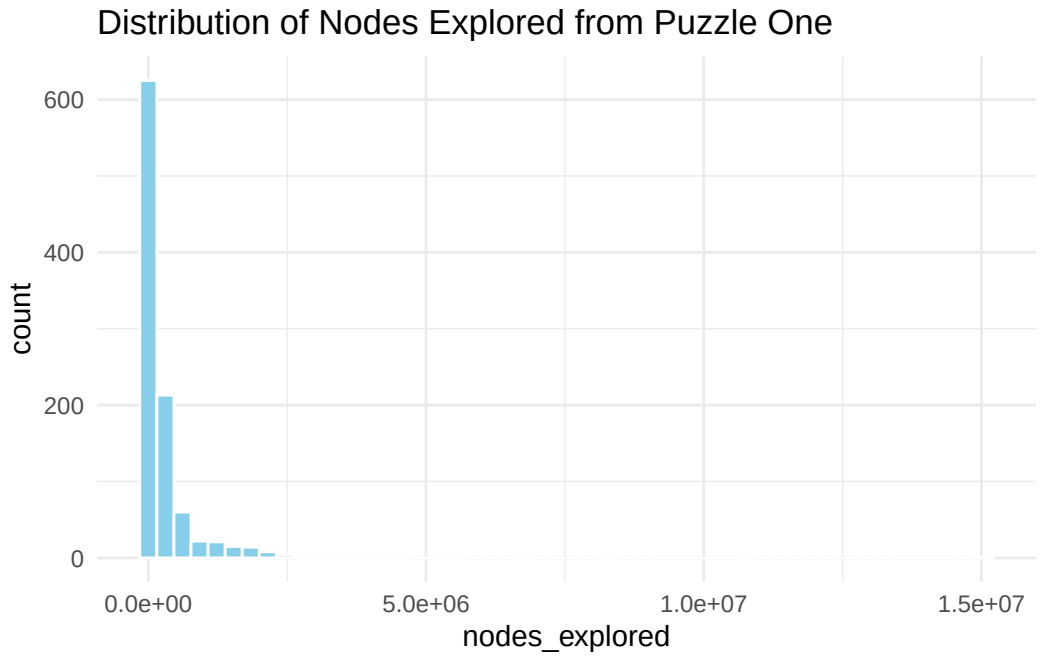
```
rand_1 <- read_csv(here("data", "rand_sample_1.csv"))

processed_rand_1 <- rand_1 |>
  filter(is_solved == TRUE)

glimpse(processed_rand_1)
```

```
Rows: 1,000
Columns: 12
$ puzzle_id           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ puzzle              <chr> "1..5.37..6.3..8.9......98...1.......8761.........~
$ clues               <dbl> 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27~
$ difficulty          <dbl> 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, ~
$ run_id              <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
$ solutions_found     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ nodes_explored      <dbl> 64805, 698410, 106409, 321338, 158532, 236683, 590~
$ max_recursion_depth <dbl> 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53~
$ solve_time_ms       <dbl> 8, 94, 14, 43, 22, 32, 0, 0, 545, 6, 1, 0, 26, 12,~
$ is_solved           <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TR~
$ leaves              <dbl> 18900, 229293, 38121, 111692, 41730, 83737, 1743, ~
$ backtracks          <dbl> 64751, 698356, 106355, 321284, 158478, 236629, 584~
```

One Random Puzzle

This plot shows the distribution of nodes explored for one Sudoku puzzle with random orderings. An ordering refers to the order in which the brute-force solver fills in the empty cells. Here is a general look at the distribution:

```
plot1 <- ggplot(processed_rand_1, aes(x = nodes_explored)) +
  geom_histogram(bins = 50, fill = "skyblue", color = "white") +
  labs(title = "Distribution of Nodes Explored from Puzzle One") +
  theme_minimal()

one_mean <- mean(processed_rand_1$nodes_explored, na.rm = TRUE)
one_median <- median(processed_rand_1$nodes_explored, na.rm = TRUE)

plot1
```




```
cat("Mean: ", one_mean, "\n")
```

```
Mean:  343673.3
```

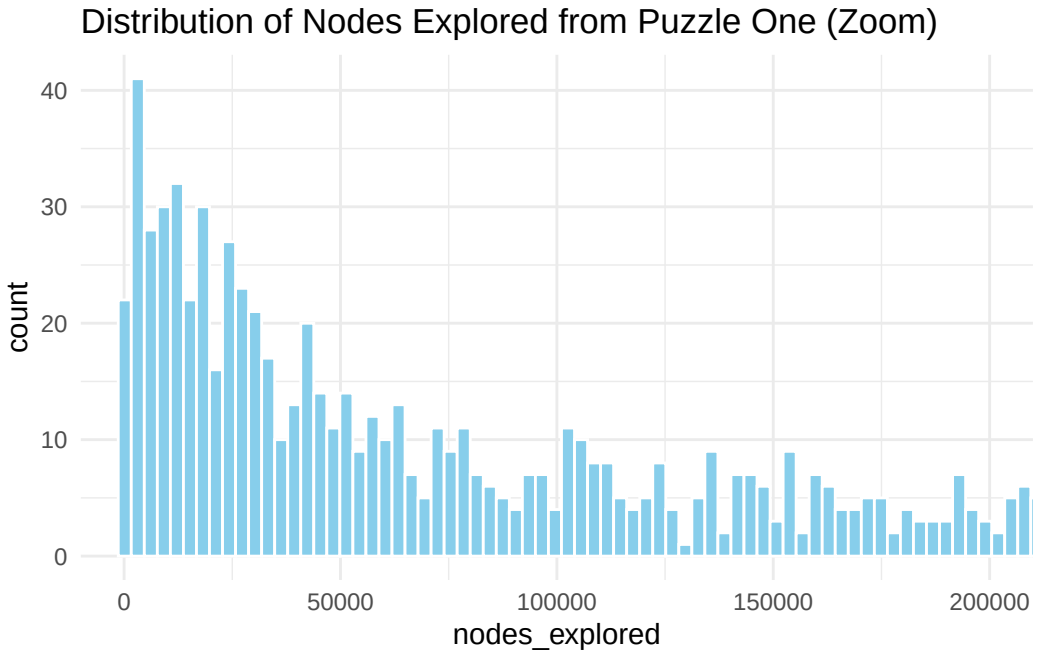
```
cat("Median: ", one_median)
```

```
Median:  91377.5
```

This distribution is extremely right skewed, indicating that for this particular puzzle, there are a few random orderings that result in extraordinary large search trees in comparison to most of the orderings.

Lets see what happens when we zoom in and examine the data points on the left. We can do this by limiting the max value of the x value to 200000.

```
plot1_zoom <- ggplot(processed_rand_1, aes(x = nodes_explored)) +
  geom_histogram(bins = 5000, fill = "skyblue", color = "white") +
  coord_cartesian(xlim = c(0, 200000)) +
  labs(title = "Distribution of Nodes Explored from Puzzle One (Zoom)") +
  theme_minimal()

plot1_zoom
```



Of course, the data is still right-skewed with most of the orderings being under 100000 nodes explored.

Questions: 1) Where is the default ordering in this distribution? 2) Does this vary per puzzle?

Another Random Puzzle

This is another random puzzle taken from the 3 million Sudoku puzzle dataset.

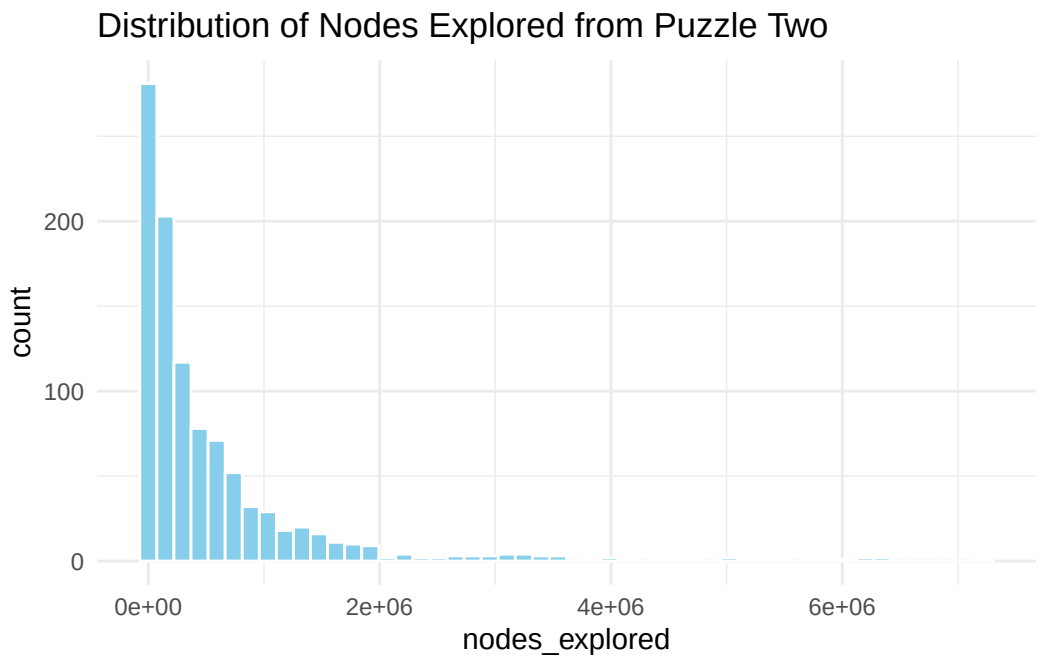
Read Data:

```
rand_2 <- read_csv(here("data", "rand_sample_1_2.csv"))

processed_rand_2 <- rand_2 |>
  filter(is_solved == TRUE)

glimpse(processed_rand_2)
```

```
Rows: 1,000
Columns: 12
$ puzzle_id           <dbl> 1358584, 1358584, 1358584, 1358584, 1358584, 13585~
$ puzzle              <chr> "47.....138....4..6.5..7.4..9362..........76...2.1~
$ clues               <dbl> 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24, 24~
$ difficulty          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ run_id              <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
$ solutions_found     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ nodes_explored      <dbl> 255820, 466895, 255548, 706707, 3759, 6381, 124667~
$ max_recursion_depth <dbl> 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56, 56~
$ solve_time_ms       <dbl> 33, 63, 34, 97, 0, 0, 167, 103, 27, 76, 2, 1, 2, 6~
$ is_solved           <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TR~
$ leaves              <dbl> 77859, 150669, 83706, 238987, 1220, 1697, 368461, ~
$ backtracks          <dbl> 255763, 466838, 255491, 706650, 3702, 6324, 124662~
```

Let's do a general plot of the distribution of the size of the search tree.

```
plot2 <- ggplot(processed_rand_2, aes(x = nodes_explored)) +
  geom_histogram(bins = 50, fill = "skyblue", color = "white") +
  labs(title = "Distribution of Nodes Explored from Puzzle Two") +
  theme_minimal()

one_mean <- mean(processed_rand_2$nodes_explored, na.rm = TRUE)
one_median <- median(processed_rand_2$nodes_explored, na.rm = TRUE)

plot2
```




```
cat("Mean: ", one_mean, "\n")
```

```
Mean:  564664.2
```

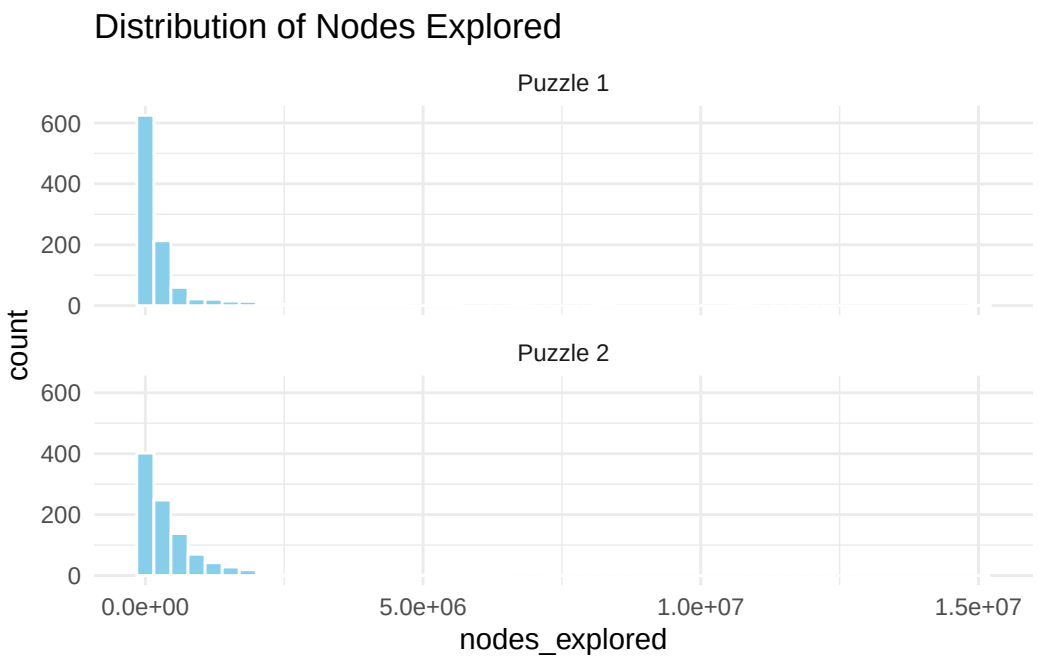
```
cat("Median: ", one_median)
```

```
Median:  233770.5
```

The mean and median of puzzle 2 are much higher than those from puzzle 1. This indicates that while they both have the same skew, perhaps the orderings from puzzle 2 are more spread out. To take a closer look lets stack them on top of each other:

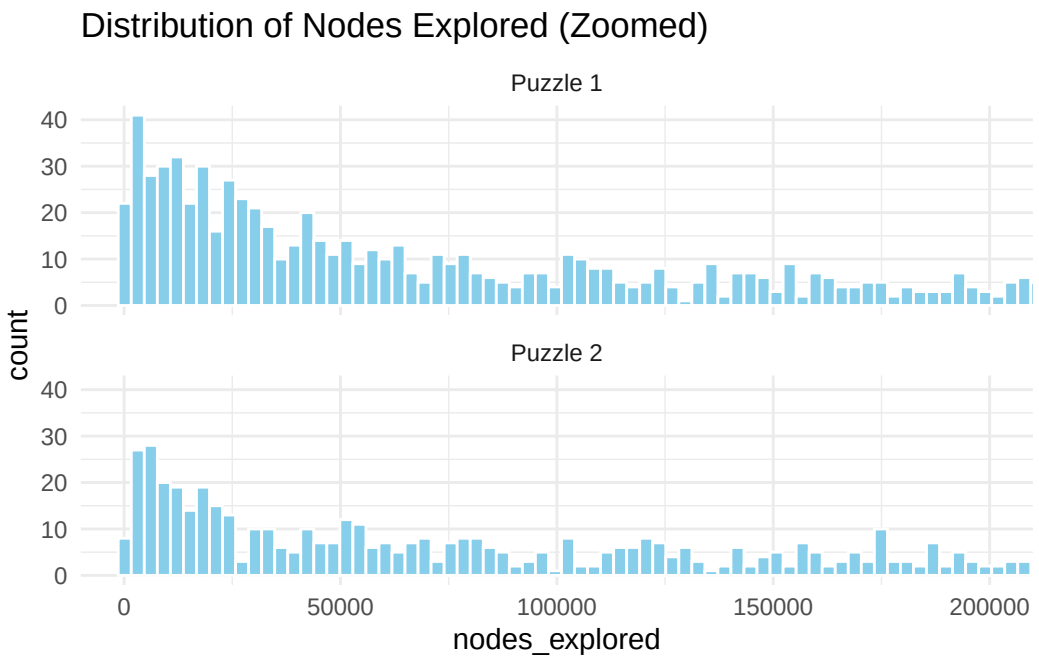
```
# Combine both datasets
combined_data <- bind_rows(
  processed_rand_1 |> mutate(puzzle = "Puzzle 1"),
  processed_rand_2 |> mutate(puzzle = "Puzzle 2")
)

# Create faceted plot - automatically shares scales
ggplot(combined_data, aes(x = nodes_explored)) +
  geom_histogram(bins = 50, fill = "skyblue", color = "white") +
```

```
  facet_wrap(~ puzzle, ncol = 1, scales = "fixed") +  # or use "fixed" for identical scales
  labs(title = "Distribution of Nodes Explored") +
  theme_minimal()
```



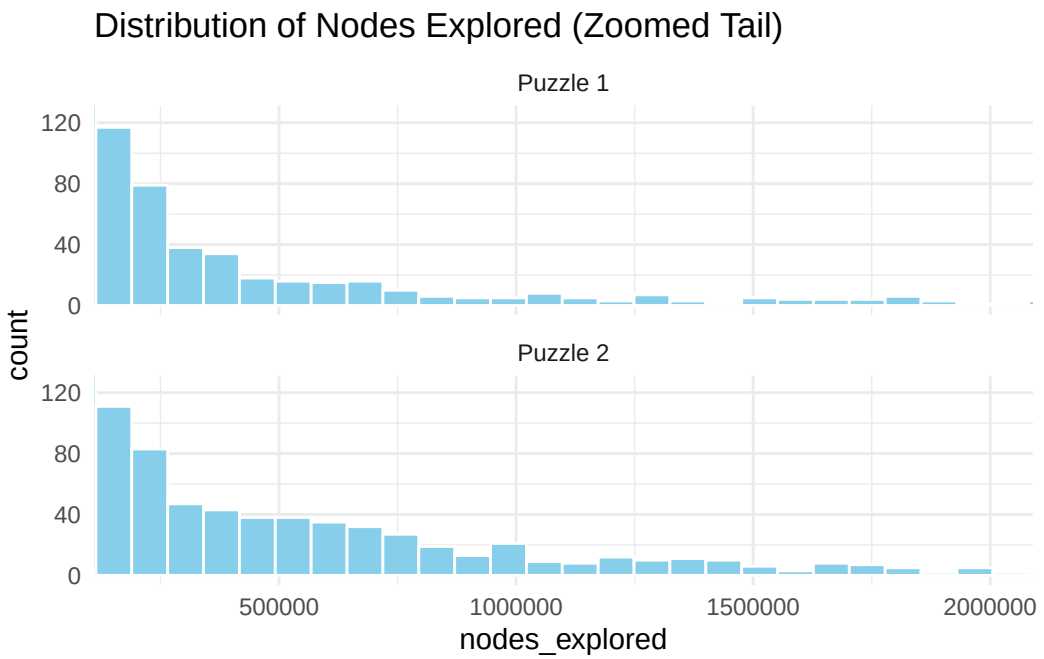
This shows that puzzle 2 has a lower proportion of orderings resulting in a smaller search tree than puzzle 1, supporting our earlier conjecture that puzzle 2 was more spread out. Similar to how we zoomed in on puzzle one's distribution, lets do the same zoom:

```
# You might want to adjust bins when zooming
ggplot(combined_data, aes(x = nodes_explored)) +
  geom_histogram(bins = 5000, fill = "skyblue", color = "white") + # More bins when zoomed
  facet_wrap(~ puzzle, ncol = 1, scales = "fixed") +
  labs(title = "Distribution of Nodes Explored (Zoomed)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 200000))
```



This zooming in gives us only a little more insight into the spread of each graph. They both have the same general shape and the only main difference is that there are just less data points on the left for puzzle 2. Thus, one would assume that there are more data points spread along the tail for puzzle 2 than for puzzle 1. Let's take a look at that in our next chart:

```
# You might want to adjust bins when zooming
ggplot(combined_data, aes(x = nodes_explored)) +
  geom_histogram(bins = 200, fill = "skyblue", color = "white") + # More bins when zoomed
  facet_wrap(~ puzzle, ncol = 1, scales = "fixed") +
  labs(title = "Distribution of Nodes Explored (Zoomed Tail)") +
  theme_minimal() +
  coord_cartesian(xlim = c(200000, 2000000),
                  ylim = c(0, 125))
```



Although its slight, we can see that there are more trees along the tail of puzzle 2 than puzzle 1.

One final thing we can do to analyze these puzzles is compare their pre-defined difficulty scores:

```
puz_1_diff <- processed_rand_1 |>
  summarize(mean_diff = mean(difficulty)) |>
  pull(mean_diff)

puz_2_diff <- processed_rand_2 |>
  summarize(mean_diff = mean(difficulty)) |>
  pull(mean_diff)

cat("Puzzle 1 Difficulty: ", puz_1_diff, "\n")
```

```
Puzzle 1 Difficulty:  2.2
```

```
cat("Puzzle 2 Difficulty: ", puz_2_diff)
```

```
Puzzle 2 Difficulty:  0
```

This would indicate that our brute force solver does not correlate well with puzzle difficulty as our solving algorithm performed worse on the second puzzle despite it having a lower difficulty score.

Modified One Sample Puzzles

This section goes over the plots generated when I modified my code to include the default ordering as the first run of each dataset. Let's do the same comparisons as before using this new data.

Read Data:

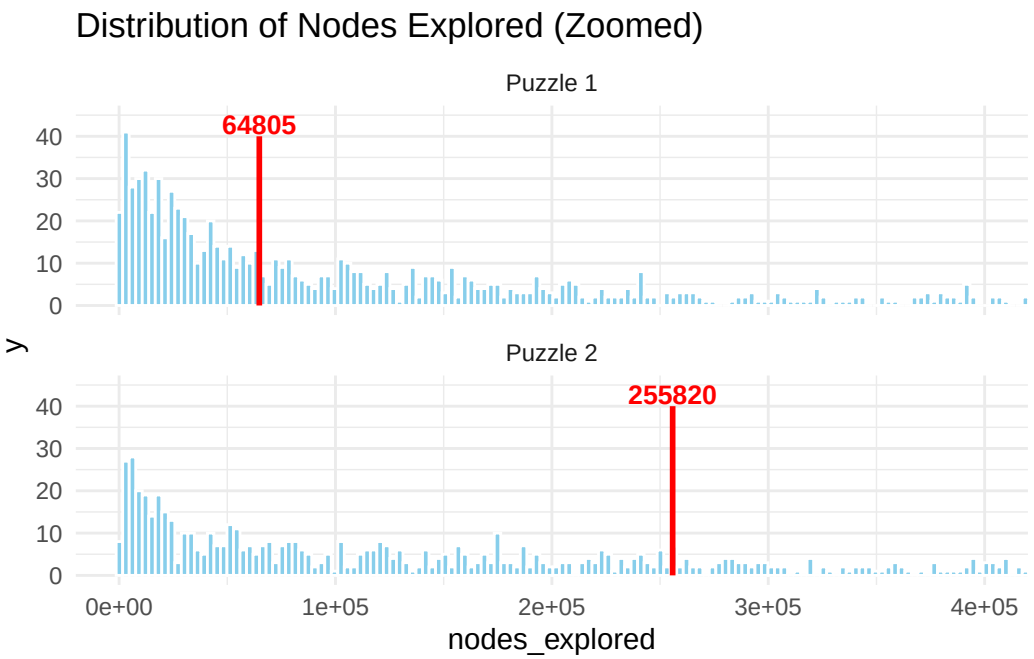
```
rand_1_mod <- read_csv(here("data", "rand_sample_1_1_mod.csv"))
rand_2_mod <- read_csv(here("data", "rand_sample_1_2_mod.csv"))

data_rand_1_mod <- rand_1 |>
  filter(is_solved == TRUE)

data_rand_2_mod <- rand_2 |>
  filter(is_solved == TRUE)

# Combine both datasets
combined_data_mod <- bind_rows(
  data_rand_1_mod |> mutate(puzzle = "Puzzle 1"),
  data_rand_2_mod |> mutate(puzzle = "Puzzle 2")
)

def_ord_values_mod <- combined_data_mod |>
  filter(run_id == 0)

glimpse(combined_data_mod)
```

```
Rows: 2,000
Columns: 12
$ puzzle_id           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ puzzle              <chr> "Puzzle 1", "Puzzle 1", "Puzzle 1", "Puzzle 1", "P~
$ clues               <dbl> 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27~
$ difficulty          <dbl> 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, 2.2, ~
$ run_id              <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
$ solutions_found     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ nodes_explored      <dbl> 64805, 698410, 106409, 321338, 158532, 236683, 590~
$ max_recursion_depth <dbl> 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53, 53~
```

```
$ solve_time_ms       <dbl> 8, 94, 14, 43, 22, 32, 0, 0, 545, 6, 1, 0, 26, 12,~
$ is_solved           <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TR~
$ leaves              <dbl> 18900, 229293, 38121, 111692, 41730, 83737, 1743, ~
$ backtracks          <dbl> 64751, 698356, 106355, 321284, 158478, 236629, 584~
```

Let's take a look at their general distributions again:

```r
# You might want to adjust bins when zooming
ggplot(combined_data, aes(x = nodes_explored)) +
  geom_histogram(bins = 5000, fill = "skyblue", color = "white") + # More bins when zoomed
  geom_segment(data = def_ord_values_mod,
               aes(x = nodes_explored, xend = nodes_explored,
                   y = 0, yend = 40),  # Customize yend to control line height
               color = "red",
               linewidth = 1,
               linetype = "solid") +
  geom_text(data = def_ord_values_mod,
            aes(x = nodes_explored, y = 45,
                label = paste(nodes_explored)),
            color = "red",
            vjust = 1,
            hjust = 0.5,
            size = 3.5,
            fontface = "bold") +
  facet_wrap(~ puzzle, ncol = 1, scales = "fixed") +
  labs(title = "Distribution of Nodes Explored (Zoomed)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 400000))
```


Distribution of Nodes Explored (Zoomed)
Puzzle 1
64805
Puzzle 2
255820
y
40
30
20
10
0
0e+00
1e+05
2e+05
3e+05
4e+05
nodes_explored