```r
library(Matrix)
library(lubridate)
library(tidyverse)
library("lme4")
library("optimx")
library(dplyr)
library(gridExtra)
library(ggplot2)
library(stringr)
library("tidyr")
library("tibble")
library(stats)
```

In this document, I load the training moderate data we used in the benchmark paper. We check the proportions of set size played in each grade level. It does not consider structure or repetitions etc. This was the first plots we did fot the development paper to see if we can consider set size as a proxy for capacity.

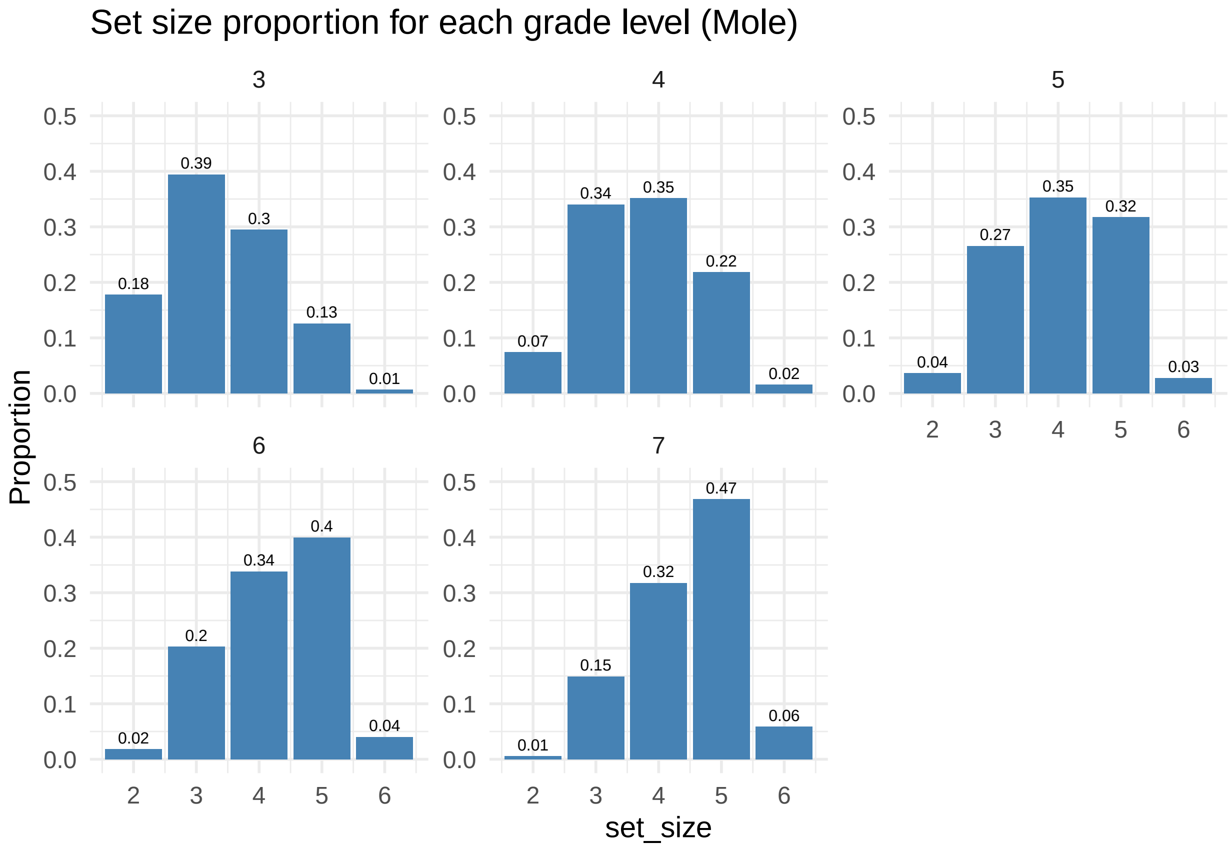
# Load Data

```r
train_logs_40_mdr <- readRDS(file = "/home/user-047/code_seyma/WMBenchmark/data/train_logs_40_mdr.rds")
train_logs_66_mdr <- readRDS(file = "/home/user-047/code_seyma/WMBenchmark/data/train_logs_66_mdr.rds")
test_logs_40_mdr <- readRDS(file = "/home/user-047/code_seyma/WMBenchmark/data/test_logs_40_mdr.rds")
test_logs_66_mdr <- readRDS(file = "/home/user-047/code_seyma/WMBenchmark/data/test_logs_66_mdr.rds")
```

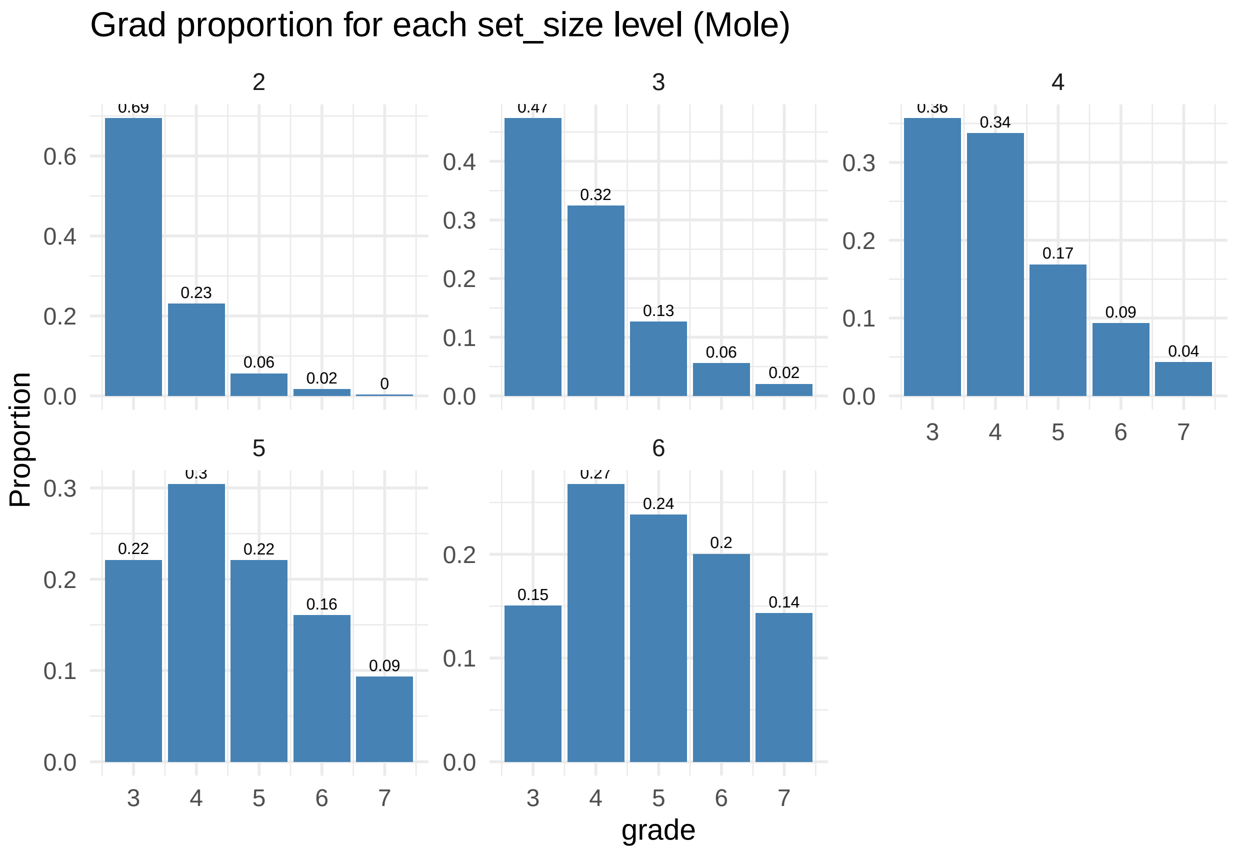
# Plots

## Mole Game

```r
# Calculate the total count for each grade
train_logs_40_mdr %>%
  group_by(grade) %>%
  summarise(total = n()) -> total_count_per_grade

# Calculate proportions
train_logs_40_mdr %>%
  group_by(grade, set_size) %>%
  count(set_size) %>%
  left_join(total_count_per_grade, by = "grade") %>%
  mutate(proportion = n / total) -> data_with_proportions

ggplot(data_with_proportions, aes(x = set_size, y = proportion)) +
  facet_wrap(~ grade, scales = "free_y") +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(
    aes(label = round(proportion, 2)),
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE,
    size = 2
  ) +
  xlab("set_size") +
  ylab("Proportion") +
  ggtitle("Set size proportion for each grade level (Mole)") +
  theme_minimal() +
  ylim(0, 0.5)
```



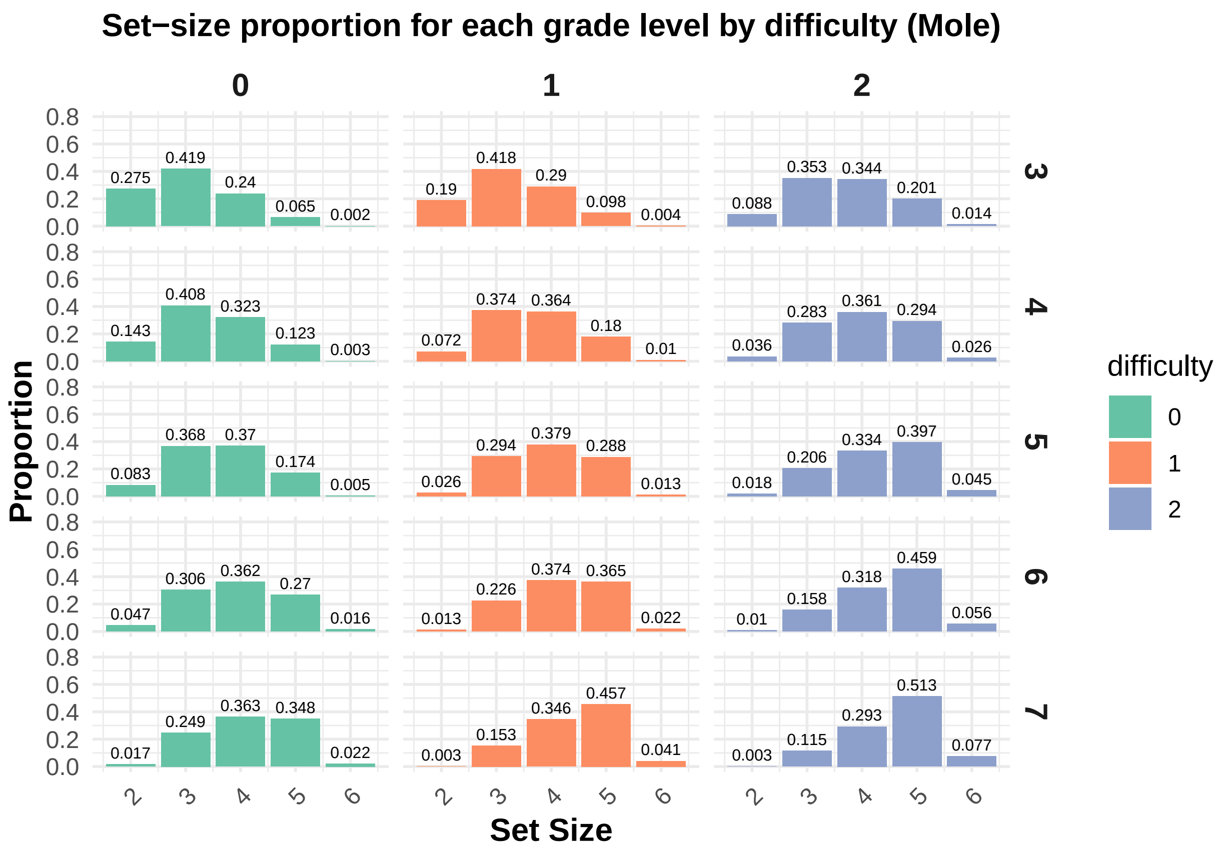
```
train_logs_40_mdr %>%
  group_by(set_size) %>%
  summarise(total = n()) -> total_count_per_setsize

train_logs_40_mdr %>%
  group_by(grade, set_size) %>%
  count(grade) %>%
  left_join(total_count_per_setsize, by = "set_size") %>%
  mutate(proportion = n / total) -> data_with_proportions

ggplot(data_with_proportions, aes(x = grade, y = proportion)) +
  facet_wrap(~ set_size, scales = "free_y") +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(
    aes(label = round(proportion, 2)), # Round to 2 decimal places
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE, # Avoid overlapping labels
    size = 2 # Adjust size as needed for readability
  ) +
  xlab("grade") +
  ylab("Proportion") +
  ggtitle("Grad proportion for each set_size level (Mole)") +
  theme_minimal()
```



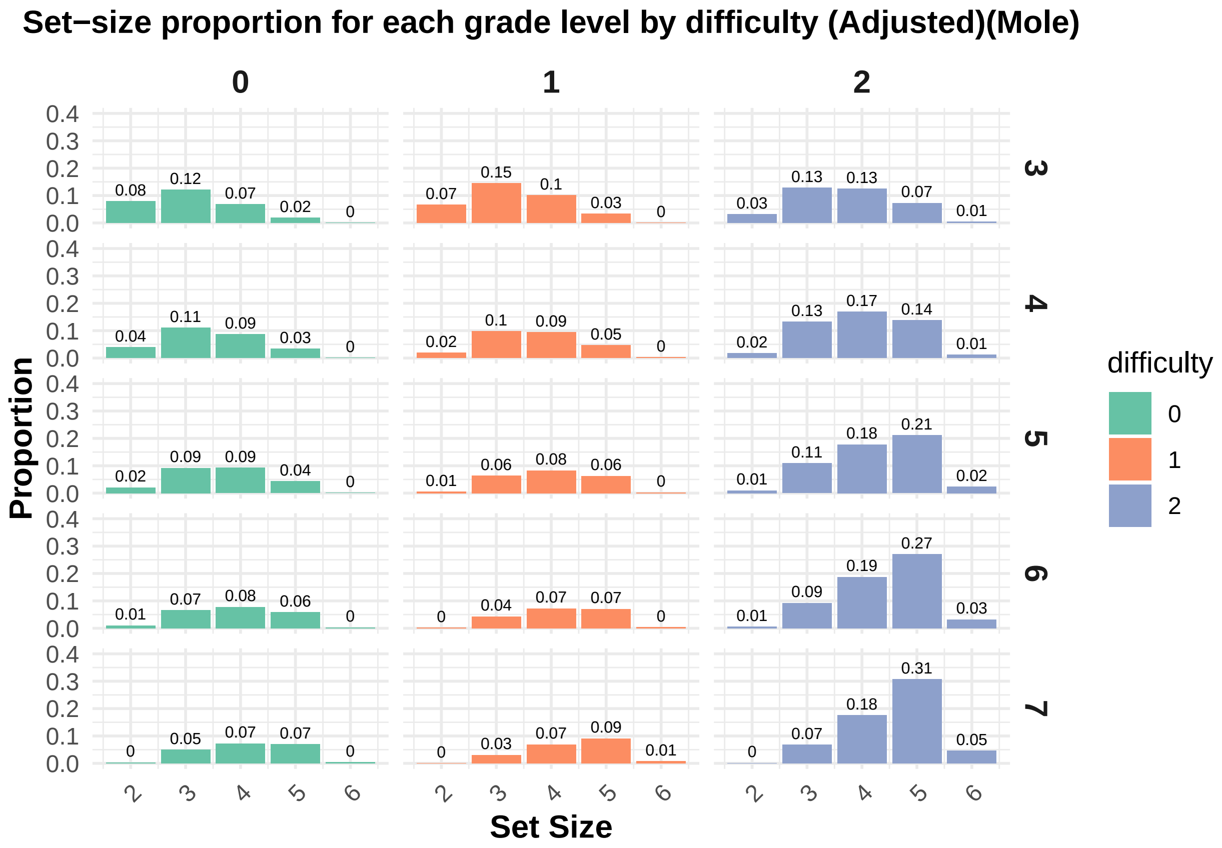
```
# Calculate the total count for each combination of grade and difficulty
train_logs_40_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade, difficulty) %>%
  summarise(total = n()) -> total_count_per_grade_difficulty
```

```
## ‘summarise()‘ has grouped output by ’grade’. You can override using the
## ‘.groups‘ argument.
```

```
# Calculate proportions
train_logs_40_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade, set_size, difficulty) %>%
  count(set_size) %>%
  left_join(total_count_per_grade_difficulty, by = c("grade", "difficulty")) %>%
  mutate(proportion = n / total) -> data_with_proportions

ggplot(data_with_proportions, aes(x = set_size, y = proportion)) +
  facet_grid(grade ~ difficulty, scales = "free_y", space = "free") +
  geom_bar(stat = "identity", aes(fill = difficulty), position = "dodge") +
    geom_text(
    aes(label = round(proportion, 3)),
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE,
    size = 2
  ) +
  scale_fill_brewer(palette = "Set2") +
  xlab("Set Size") +
  ylab("Proportion") +
  ggtitle("Set-size proportion for each grade level by difficulty (Mole)") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    strip.background = element_blank(),
    strip.text = element_text(size = 12, face = "bold")
  ) +
  ylim(0, 0.8)
```



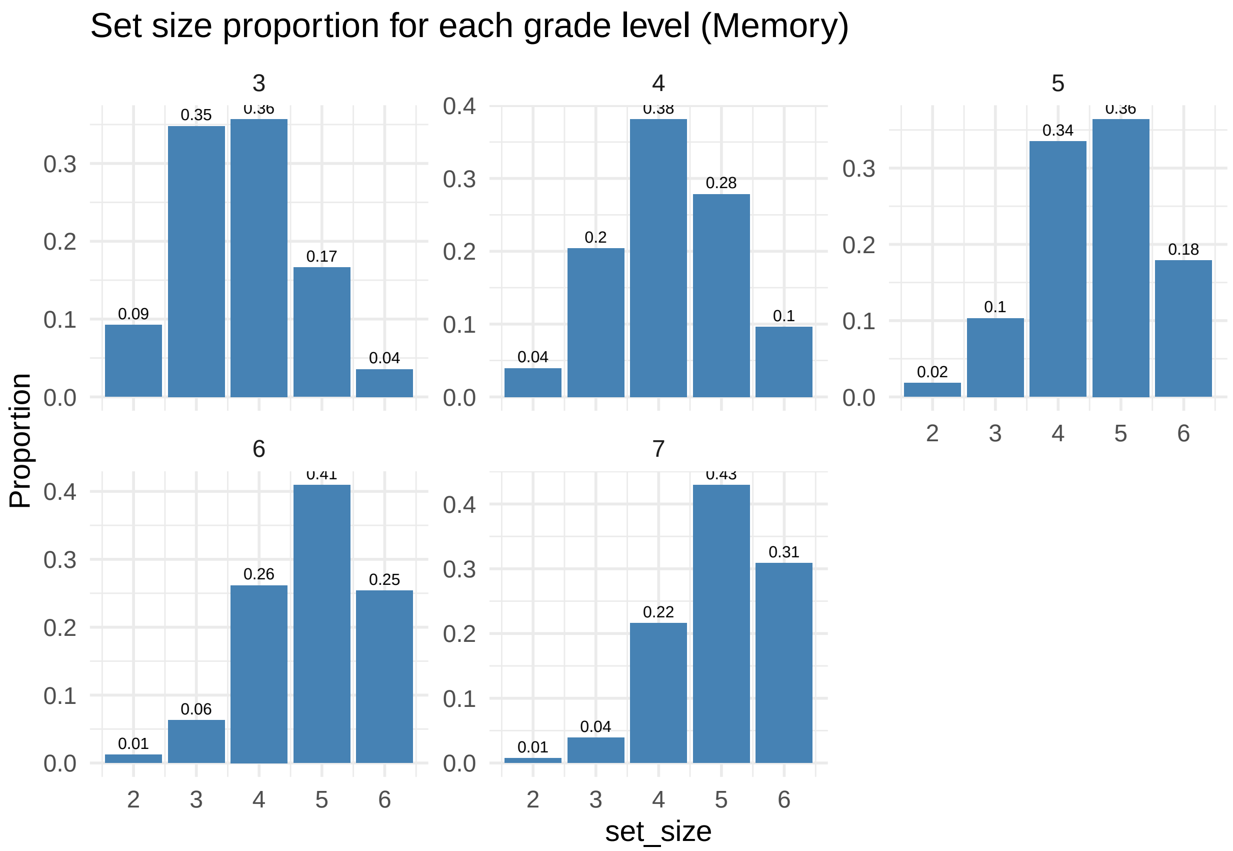
```
train_logs_40_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade) %>%
  summarise(total = n()) -> total_count_per_grade
train_logs_40_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade, set_size, difficulty) %>%
  count(set_size) %>%
  left_join(total_count_per_grade, by = "grade") %>%
  mutate(proportion = n / total) -> data_with_proportions
ggplot(data_with_proportions, aes(x = set_size, y = proportion)) +
  facet_grid(grade ~ difficulty, scales = "free_y", space = "free") +
  geom_bar(stat = "identity", aes(fill = difficulty), position = "dodge") +
  geom_text(
    aes(label = round(proportion, 2)),
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE,
    size = 2
  ) +
  scale_fill_brewer(palette = "Set2") +
  xlab("Set Size") +
  ylab("Proportion") +
  ggtitle("Set-size proportion for each grade level by difficulty (Adjusted)(Mole)") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    strip.background = element_blank(),
    strip.text = element_text(size = 12, face = "bold")
  ) +
  ylim(0, 0.4)
```



## Number Game

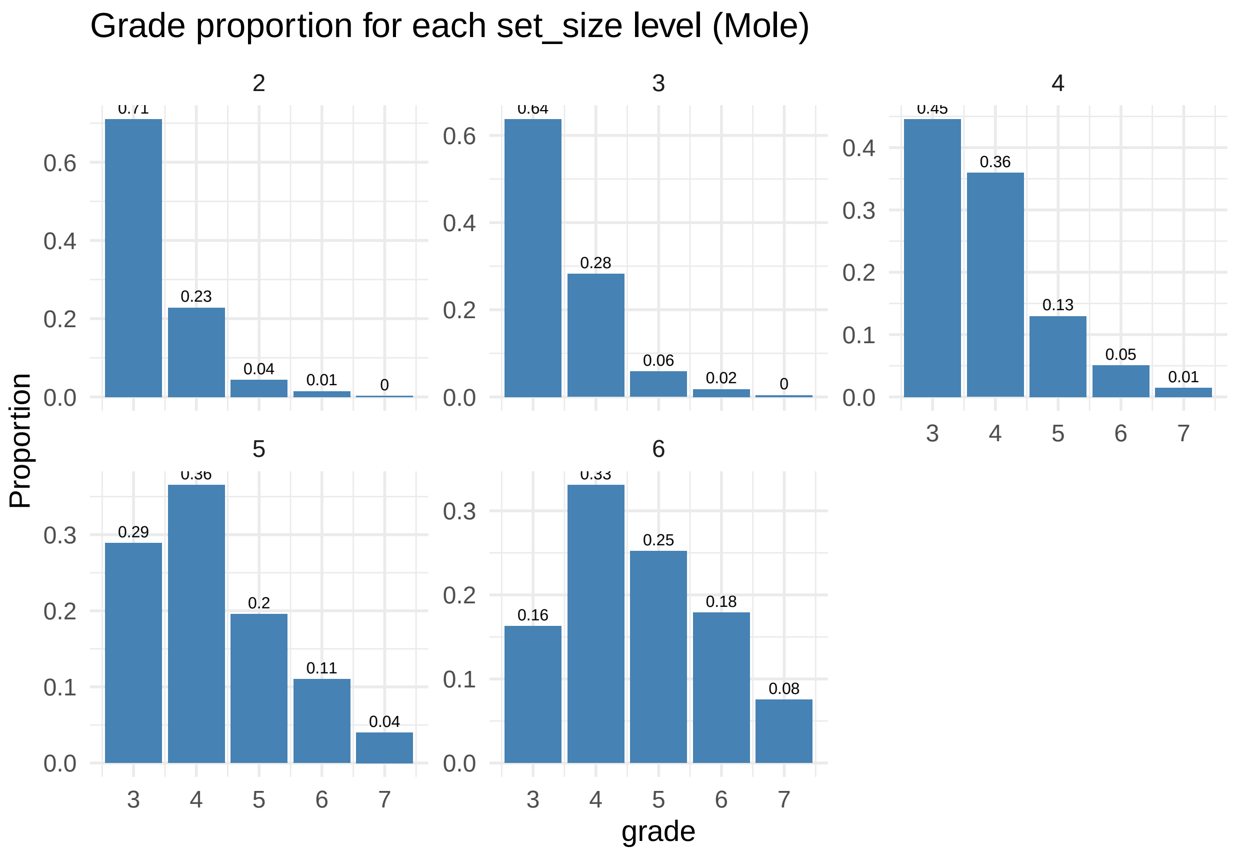
```
# Calculate the total count for each grade
train_logs_66_mdr %>%
  group_by(grade) %>%
  summarise(total = n()) -> total_count_per_grade

# Calculate proportions
train_logs_66_mdr %>%
  group_by(grade, set_size) %>%
  count(set_size) %>%
  left_join(total_count_per_grade, by = "grade") %>%
  mutate(proportion = n / total) -> data_with_proportions
```

```r
ggplot(data_with_proportions, aes(x = set_size, y = proportion)) +
  facet_wrap(~ grade, scales = "free_y") +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(
    aes(label = round(proportion, 2)),
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE,
    size = 2
  ) +
  xlab("set_size") +
  ylab("Proportion") +
  ggtitle("Set size proportion for each grade level (Memory)") +
  theme_minimal()
```



```r
train_logs_66_mdr %>%
  group_by(set_size) %>%
  summarise(total = n()) -> total_count_per_setsize

train_logs_66_mdr %>%
  group_by(grade, set_size) %>%
  count(grade) %>%
  left_join(total_count_per_setsize, by = "set_size") %>%
  mutate(proportion = n / total) -> data_with_proportions
```

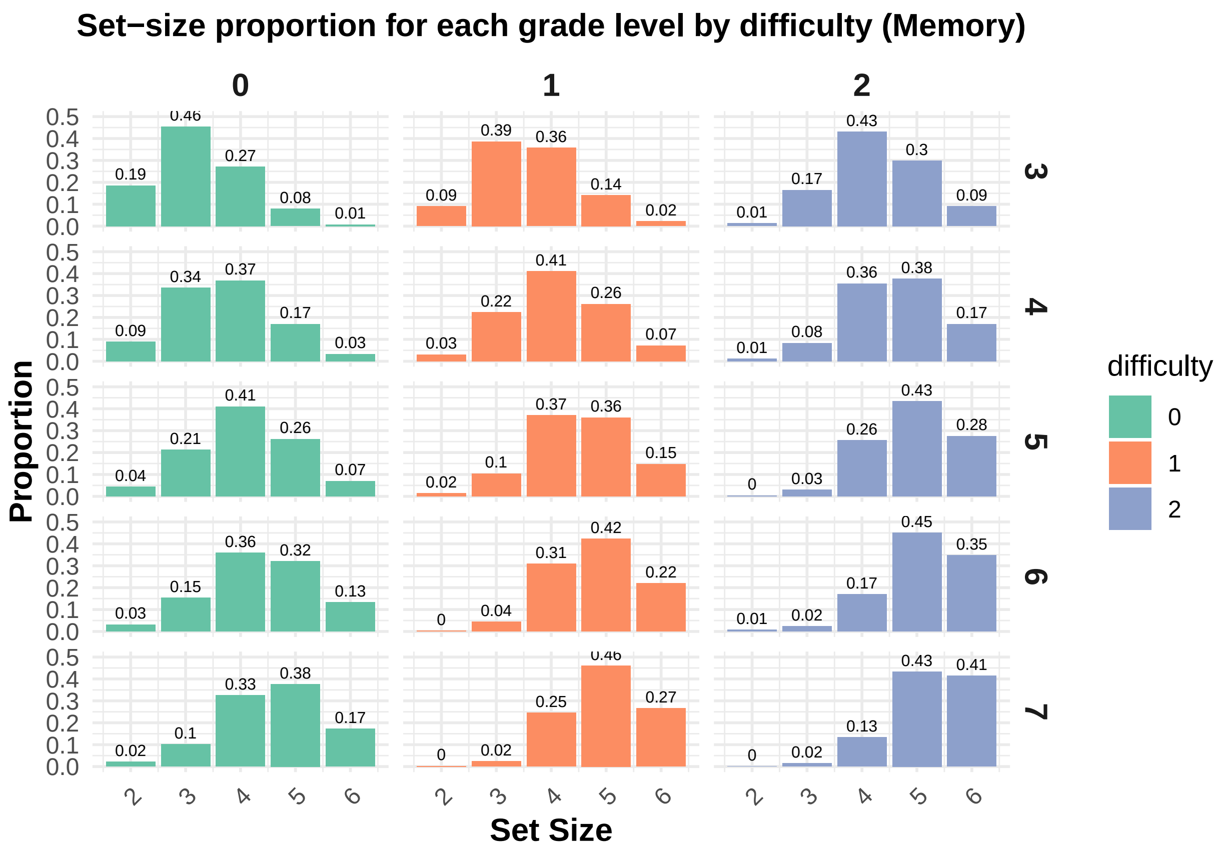
```r
ggplot(data_with_proportions, aes(x = grade, y = proportion)) +
  facet_wrap(~ set_size, scales = "free_y") +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(
    aes(label = round(proportion, 2)),
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE,
    size = 2
  ) +
  xlab("grade") +
  ylab("Proportion") +
  ggtitle("Grade proportion for each set_size level (Mole)") +
  theme_minimal()
```



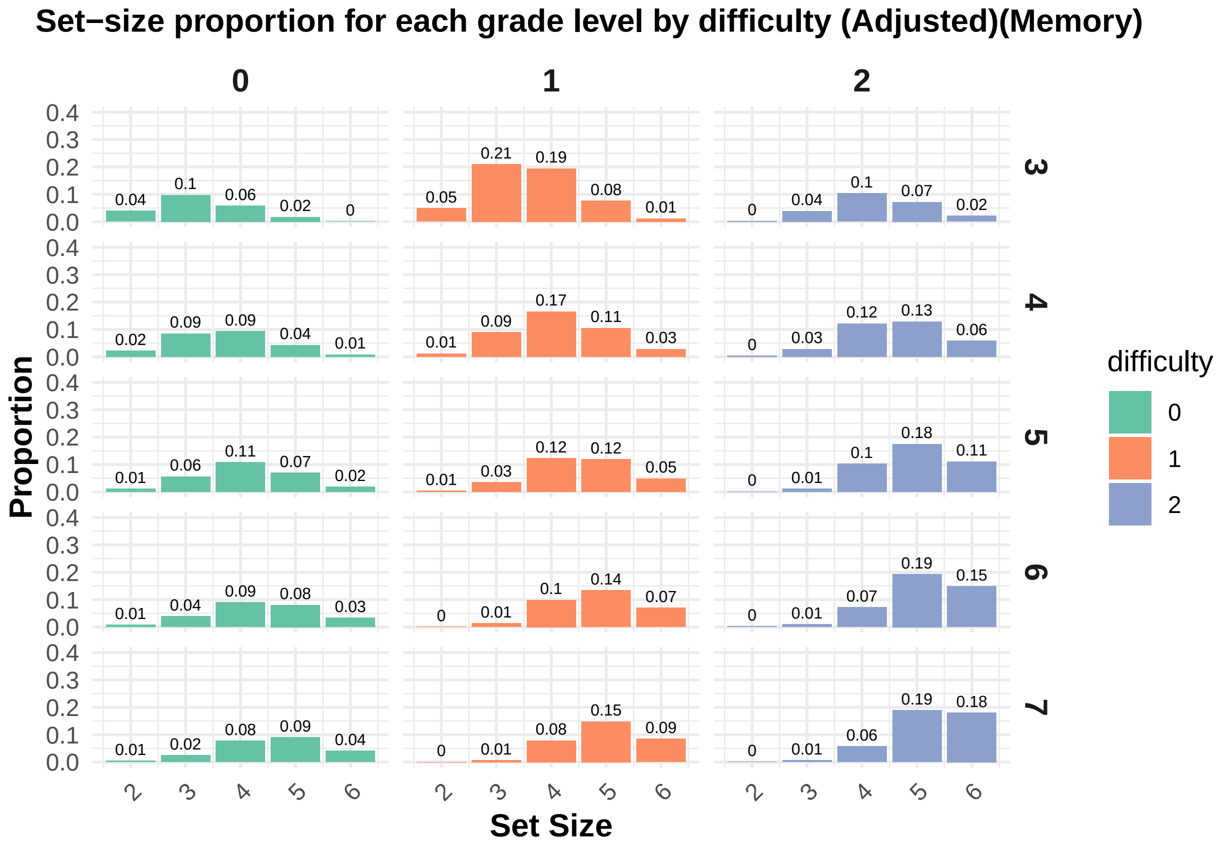
```r
# within each difficulty level for each grade.
train_logs_66_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade, difficulty) %>%
  summarise(total = n()) -> total_count_per_grade_difficulty
```

```
## ‘summarise()‘ has grouped output by ’grade’. You can override using the
## ‘.groups‘ argument.
```

```r
train_logs_66_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade, set_size, difficulty) %>%
  count(set_size) %>%
  left_join(total_count_per_grade_difficulty, by = c("grade", "difficulty")) %>%
  mutate(proportion = n / total) -> data_with_proportions

#Each difficulty panel is scaled separately within grade
#istribution within difficulty levels
ggplot(data_with_proportions, aes(x = set_size, y = proportion, fill = difficulty)) +
  facet_grid(grade ~ difficulty, scales = "free_y", space = "free") +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  geom_text(
    aes(label = round(proportion, 2)), # Round to 2 decimal places
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE, # Avoid overlapping labels
    size = 2 # Adjust size as needed for readability
  ) +
  scale_fill_brewer(palette = "Set2") +
  xlab("Set Size") +
  ylab("Proportion") +
  ggtitle("Set-size proportion for each grade level by difficulty (Memory)") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 12, face = "bold", hjust = 0.5),
    strip.background = element_blank(),
    strip.text = element_text(size = 12, face = "bold")
  ) +
  ylim(0, 0.5)
```



```
train_logs_66_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade) %>%
  summarise(total = n()) -> total_count_per_grade
train_logs_66_mdr %>%
  mutate(grade = as.factor(grade), difficulty = as.factor(difficulty)) %>%
  group_by(grade, set_size, difficulty) %>%
  count(set_size) %>%
  left_join(total_count_per_grade, by = "grade") %>%
  mutate(proportion = n / total) -> data_with_proportions

#All difficulty values within the grade level adds up to 1
ggplot(data_with_proportions, aes(x = set_size, y = proportion)) +
  facet_grid(grade ~ difficulty, scales = "free_y", space = "free") +
  geom_bar(stat = "identity", aes(fill = difficulty), position = "dodge") +
  geom_text(
    aes(label = round(proportion, 2)),
    vjust = -0.5,
    position = position_dodge(width = 0.9),
    check_overlap = TRUE,
    size = 2
  ) +
  scale_fill_brewer(palette = "Set2") +
  xlab("Set Size") +
  ylab("Proportion") +
  ggtitle("Set-size proportion for each grade level by difficulty (Adjusted)(Memory)") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 12, face = "bold", hjust = 0.3),
    strip.background = element_blank(),
    strip.text = element_text(size = 12, face = "bold")
  ) +
  ylim(0, 0.4)
```


Set−size proportion for each grade level by difficulty (Adjusted)(Memory)
Proportion
Set Size
difficulty