Shiny

```
#1. Data Preparation
# Install and load necessary libraries
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2")
if (!requireNamespace("dplyr", quietly = TRUE)) install.packages("dplyr")
if (!requireNamespace("ggalluvial", quietly = TRUE)) install.packages("ggalluvial")
if (!requireNamespace("rnaturalearth", quietly = TRUE)) install.packages("rnaturalearth")
if (!requireNamespace("sf", quietly = TRUE)) install.packages("sf")
if (!requireNamespace("tidyr", quietly = TRUE)) install.packages("tidyr")
if (!requireNamespace("ggridges", quietly = TRUE)) install.packages("ggridges")
if (!requireNamespace("treemapify", quietly = TRUE)) install.packages("treemapify")
if (!requireNamespace("terra", quietly = TRUE)) install.packages("terra")

library(ggplot2)
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(tidyr)
library(ggalluvial)
library(sf)
```

```
## Linking to GEOS 3.11.2, GDAL 3.8.2, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(rnaturalearth)
library(ggridges)
library(treemapify)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(terra)
```

```
## terra 1.7.83
```

```
##
## Attaching package: 'terra'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
data<- read.csv("/Users/Leen/Downloads/FinalDataDV.csv")
head(data)
```

```
##   Marital.status                 Application.mode Application.order
## 1         Single     2nd phase-general contingent                 5
## 2         Single International student (bachelor)                 1
## 3         Single     1st phase-general contingent                 5
## 4         Single     2nd phase-general contingent                 2
## 5        Married                Over 23 years old                 1
## 6        Married                Over 23 years old                 1
##                              Course Daytime.evening.attendance
## 1     Animation and Multimedia Design                   Daytime
## 2                            Tourism                   Daytime
## 3                Communication Design                   Daytime
## 4        Journalism and Communication                   Daytime
## 5 Social Service (evening attendance)                   Evening
## 6     Management (evening attendance)                   Evening
##                                       Previous.qualification Nacionality
## 1                                        Secondary education  Portuguese
## 2                                        Secondary education  Portuguese
## 3                                        Secondary education  Portuguese
## 4                                        Secondary education  Portuguese
## 5                                        Secondary education  Portuguese
## 6 Basic education 3rd cycle (9th/10th/11th year) or equivalent  Portuguese
##   Mother.s.qualification Father.s.qualification Mother.s.occupation
## 1                     13                     10                   6
## 2                      1                      3                   4
## 3                     22                     27                  10
## 4                     23                     27                   6
## 5                     22                     28                  10
## 6                     22                     27                  10
##   Father.s.occupation Displaced Educational.special.needs Debtor
## 1                  10       Yes                       No     No
## 2                   4       Yes                       No     No
## 3                  10       Yes                       No     No
## 4                   4       Yes                       No     No
## 5                  10        No                       No     No
## 6                   8        No                       No    Yes
##   Tuition.fees.up.to.date Gender Scholarship.holder Age.at.enrollment
## 1                     Yes   Male                 No                20
## 2                      No   Male                 No                19
```

```
## 3                     No   Male                No                19
## 4                    Yes Female                No                20
## 5                    Yes Female                No                45
## 6                    Yes   Male                No                50
##   International Curricular.units.1st.sem..credited.
## 1            No                                  0
## 2            No                                  0
## 3            No                                  0
## 4            No                                  0
## 5            No                                  0
## 6            No                                  0
##   Curricular.units.1st.sem..enrolled. Curricular.units.1st.sem..evaluations.
## 1                                   0                                      0
## 2                                   6                                      6
## 3                                   6                                      0
## 4                                   6                                      8
## 5                                   6                                      9
## 6                                   5                                     10
##   Curricular.units.1st.sem..approved. Curricular.units.1st.sem..grade.
## 1                                   0                          0.00000
## 2                                   6                         14.00000
## 3                                   0                          0.00000
## 4                                   6                         13.42857
## 5                                   5                         12.33333
## 6                                   5                         11.85714
##   Curricular.units.1st.sem..without.evaluations.
## 1                                              0
## 2                                              0
## 3                                              0
## 4                                              0
## 5                                              0
## 6                                              0
##   Curricular.units.2nd.sem..credited. Curricular.units.2nd.sem..enrolled.
## 1                                   0                                   0
## 2                                   0                                   6
## 3                                   0                                   6
## 4                                   0                                   6
## 5                                   0                                   6
## 6                                   0                                   5
##   Curricular.units.2nd.sem..evaluations. Curricular.units.2nd.sem..approved.
## 1                                     0                                   0
## 2                                     6                                   6
## 3                                     0                                   0
## 4                                    10                                   5
## 5                                     6                                   6
## 6                                    17                                   5
##   Curricular.units.2nd.sem..grade.
## 1                          0.00000
## 2                         13.66667
## 3                          0.00000
## 4                         12.40000
## 5                         13.00000
## 6                         11.50000
##   Curricular.units.2nd.sem..without.evaluations. Unemployment.rate
```

```
## 1                                           0              10.8
## 2                                           0              13.9
## 3                                           0              10.8
## 4                                           0               9.4
## 5                                           0              13.9
## 6                                           5              16.2
##   Inflation.rate   GDP   Target First.Sem.Performance Second.Sem.Performance
## 1            1.4  1.74  Dropout               Failing                Failing
## 2           -0.3  0.79 Graduate                  Good                   Good
## 3            1.4  1.74  Dropout               Failing                Failing
## 4           -0.8 -3.12 Graduate                  Good                   Good
## 5           -0.3  0.79 Graduate                  Good                   Good
## 6            0.3 -0.92 Graduate               Passing                Passing
```

```
colnames(data)
```

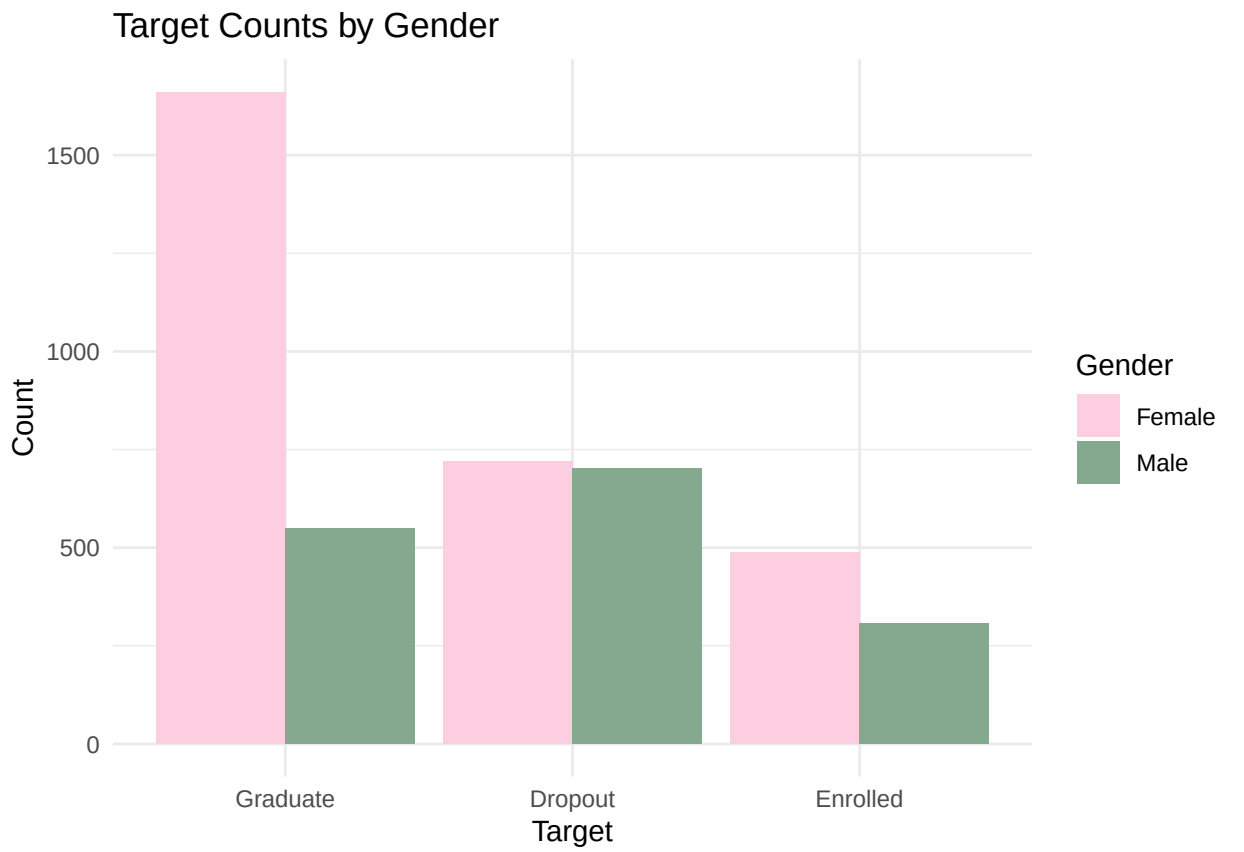
```
##  [1] "Marital.status"
##  [2] "Application.mode"
##  [3] "Application.order"
##  [4] "Course"
##  [5] "Daytime.evening.attendance"
##  [6] "Previous.qualification"
##  [7] "Nacionality"
##  [8] "Mother.s.qualification"
##  [9] "Father.s.qualification"
## [10] "Mother.s.occupation"
## [11] "Father.s.occupation"
## [12] "Displaced"
## [13] "Educational.special.needs"
## [14] "Debtor"
## [15] "Tuition.fees.up.to.date"
## [16] "Gender"
## [17] "Scholarship.holder"
## [18] "Age.at.enrollment"
## [19] "International"
## [20] "Curricular.units.1st.sem..credited."
## [21] "Curricular.units.1st.sem..enrolled."
## [22] "Curricular.units.1st.sem..evaluations."
## [23] "Curricular.units.1st.sem..approved."
## [24] "Curricular.units.1st.sem..grade."
## [25] "Curricular.units.1st.sem..without.evaluations."
## [26] "Curricular.units.2nd.sem..credited."
## [27] "Curricular.units.2nd.sem..enrolled."
## [28] "Curricular.units.2nd.sem..evaluations."
## [29] "Curricular.units.2nd.sem..approved."
## [30] "Curricular.units.2nd.sem..grade."
## [31] "Curricular.units.2nd.sem..without.evaluations."
## [32] "Unemployment.rate"
## [33] "Inflation.rate"
## [34] "GDP"
## [35] "Target"
## [36] "First.Sem.Performance"
## [37] "Second.Sem.Performance"
```

```
###
# Aggregate data
target_gender_counts <- data %>%
  group_by(Target, Gender) %>%
  summarise(Count = n()) %>%
  mutate(Percentage = Count / sum(Count) * 100)
```

```
## `summarise()` has grouped output by 'Target'. You can override using the
## `.groups` argument.
```

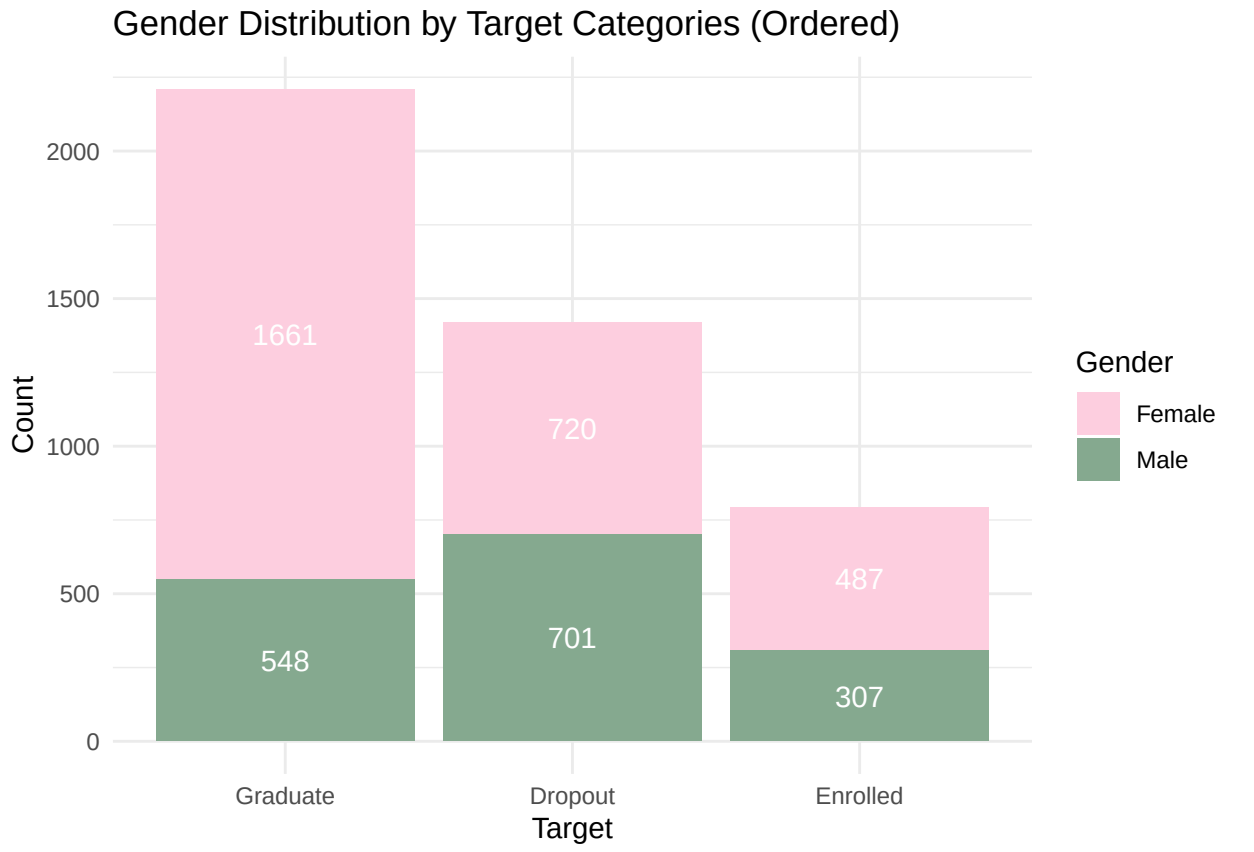
```
# Bar chart with updated legend
ggplot(target_gender_counts, aes(x = reorder(Target, -Count), y = Count, fill = as.factor(Gender))) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_discrete(labels = c("0" = "Female", "1" = "Male")) +
  scale_fill_manual(values = c("Male" = "#85A98F", "Female" = "#FDCEDF")) +
  labs(title = "Target Counts by Gender", x = "Target", y = "Count", fill = "Gender") +
  theme_minimal()
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

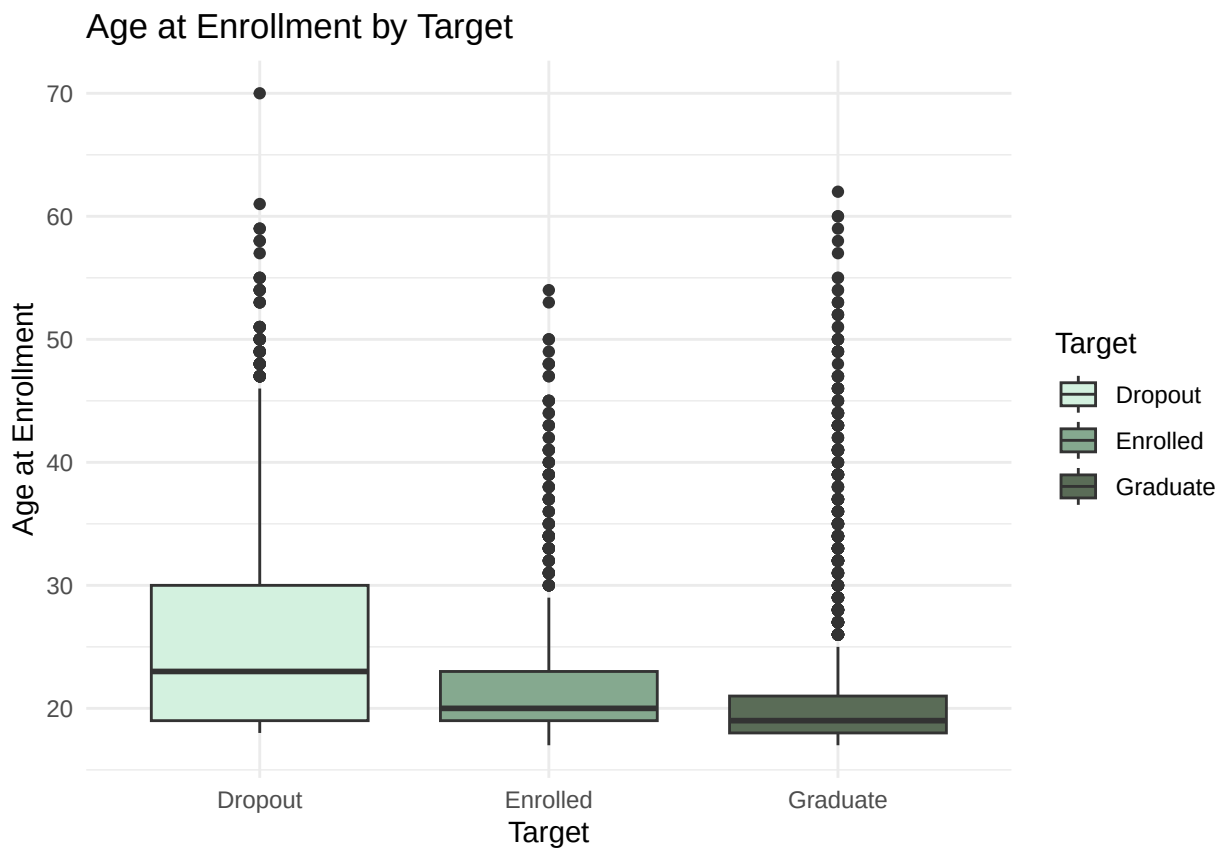



```
library(ggplot2)
library(dplyr)
library(forcats)
```

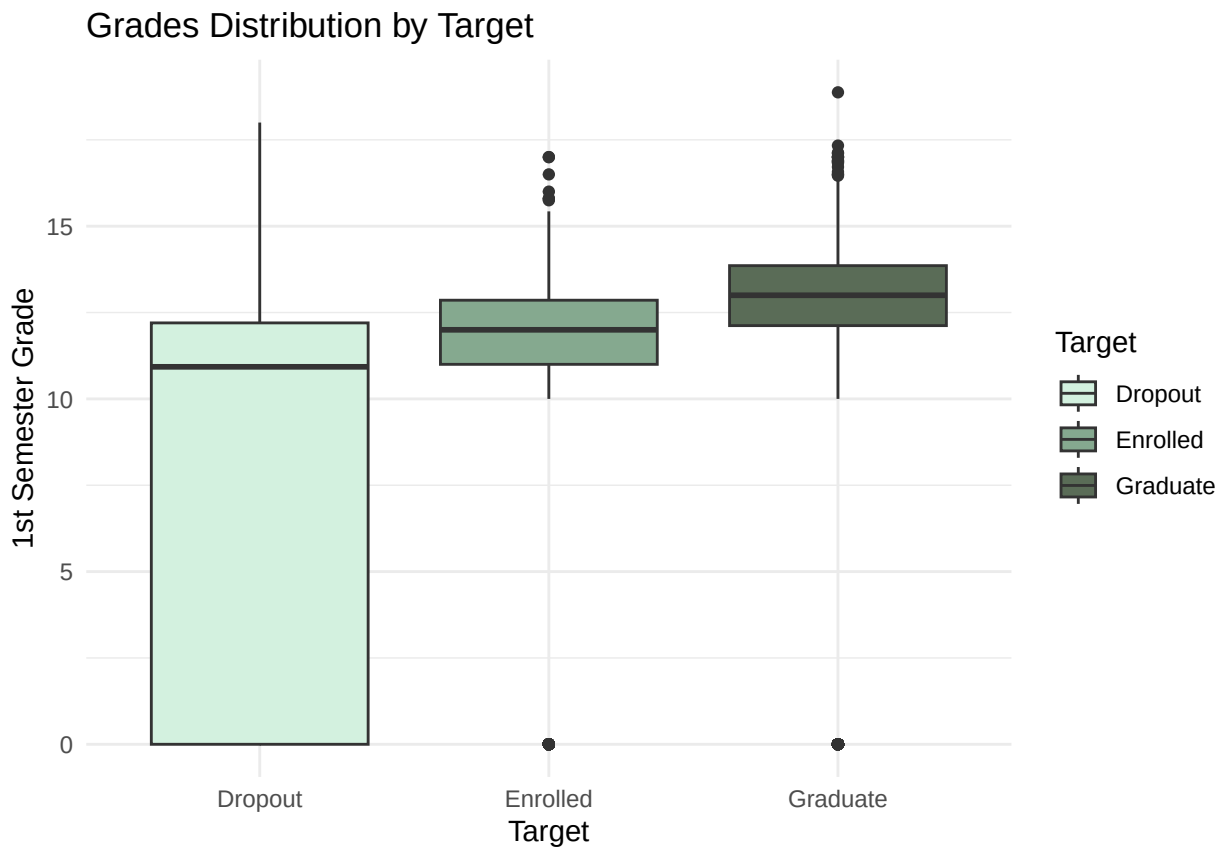
```r
# Aggregate counts for Gender by Target
gender_target_counts <- data %>%
  group_by(Target, Gender) %>%
  summarise(Count = n(), .groups = 'drop')

# Calculate total counts for each Target and reorder the Target factor
gender_target_counts <- gender_target_counts %>%
  group_by(Target) %>%
  mutate(Total = sum(Count)) %>%
  ungroup() %>%
  mutate(Target = fct_reorder(Target, Total, .desc = TRUE))

# Create the stacked bar chart
ggplot(gender_target_counts, aes(x = Target, y = Count, fill = Gender)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5), color = "white") +
  scale_fill_manual(values = c("Male" = "#85A98F", "Female" = "#FDCEDF")) +
  labs(title = "Gender Distribution by Target Categories (Ordered)",
       x = "Target",
       y = "Count",
       fill = "Gender") +
  theme_minimal()
```



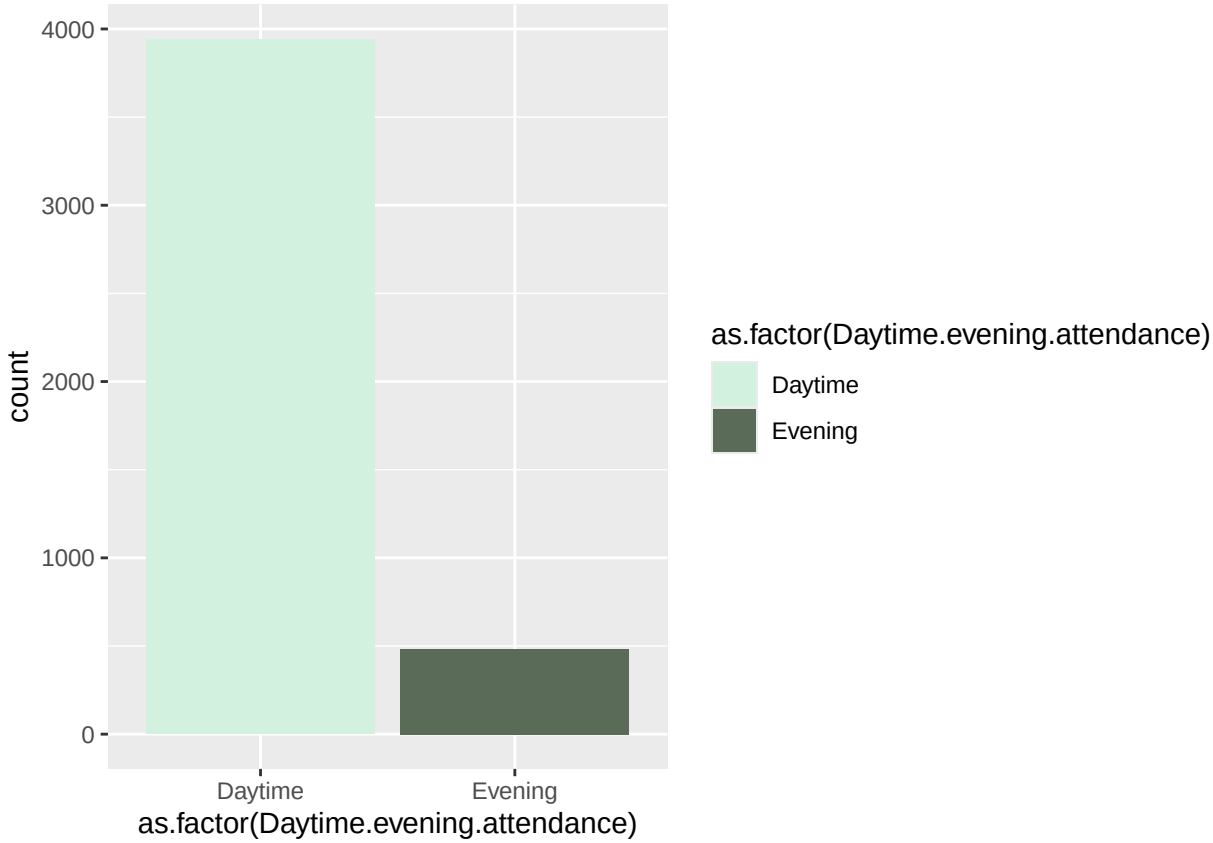
```r
# Box Plot
ggplot(data, aes(x = Target, y = Age.at.enrollment, fill = Target)) +
  geom_boxplot() +
  labs(title = "Age at Enrollment by Target", x = "Target", y = "Age at Enrollment") +
  scale_fill_manual(values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"))+
  theme_minimal()
```




```r
# Box Plot for Grades
ggplot(data, aes(x = Target, y = Curricular.units.1st.sem..grade., fill = Target)) +
  geom_boxplot() +
  labs(title = "Grades Distribution by Target", x = "Target", y = "1st Semester Grade") +
  scale_fill_manual(values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"))+
  theme_minimal()
```



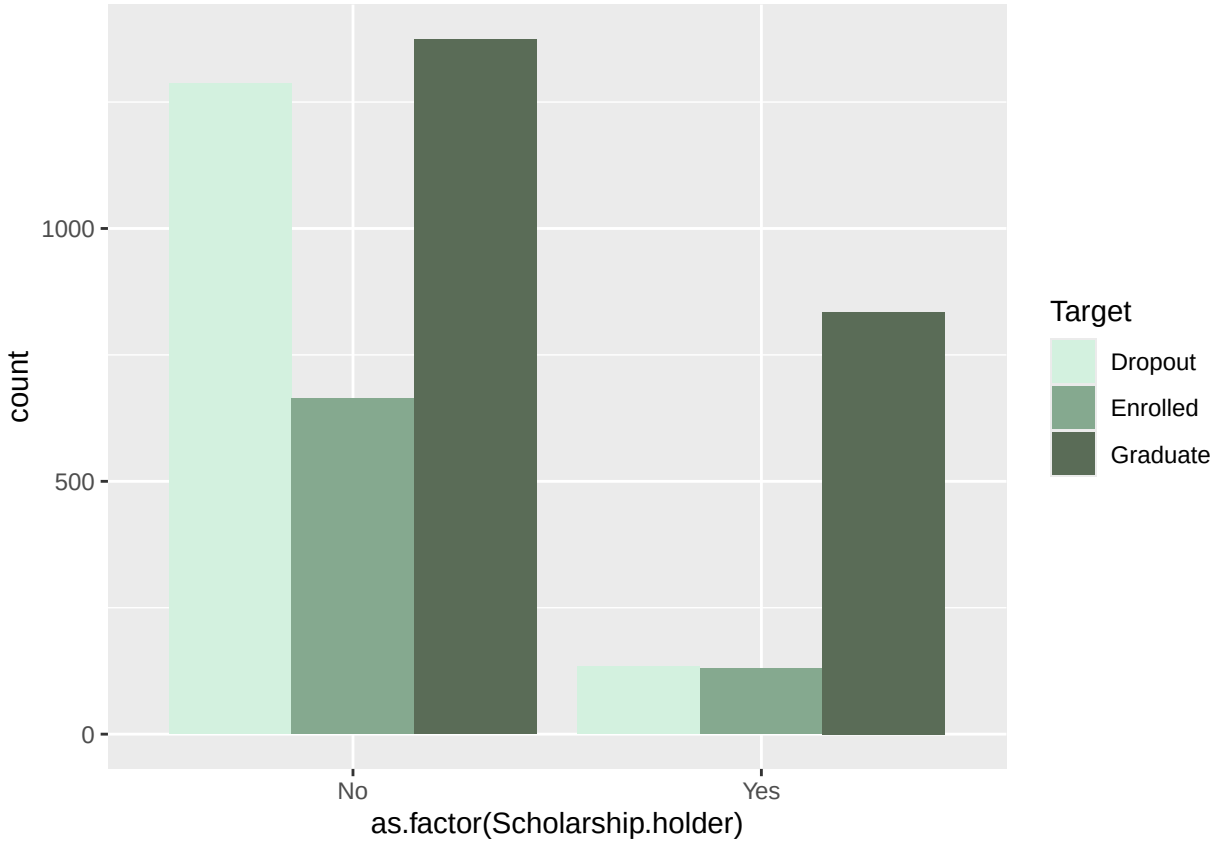

```
ggplot(data, aes(x = as.factor(Daytime.evening.attendance), fill = as.factor(Daytime.evening.attendance)
  geom_bar() +
  scale_fill_manual(values = c("Daytime" = "#D3F1DF", "Evening" = "#5A6C57"))
```




```
  labs(title = "Daytime vs Evening Attendance", x = "Attendance ", y = "Count") +
  theme_minimal()
```

```
## NULL
```

```
ggplot(data, aes(x = as.factor(Scholarship.holder), fill = Target)) +
  geom_bar(position = "dodge") +
   scale_fill_manual(values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"))
```

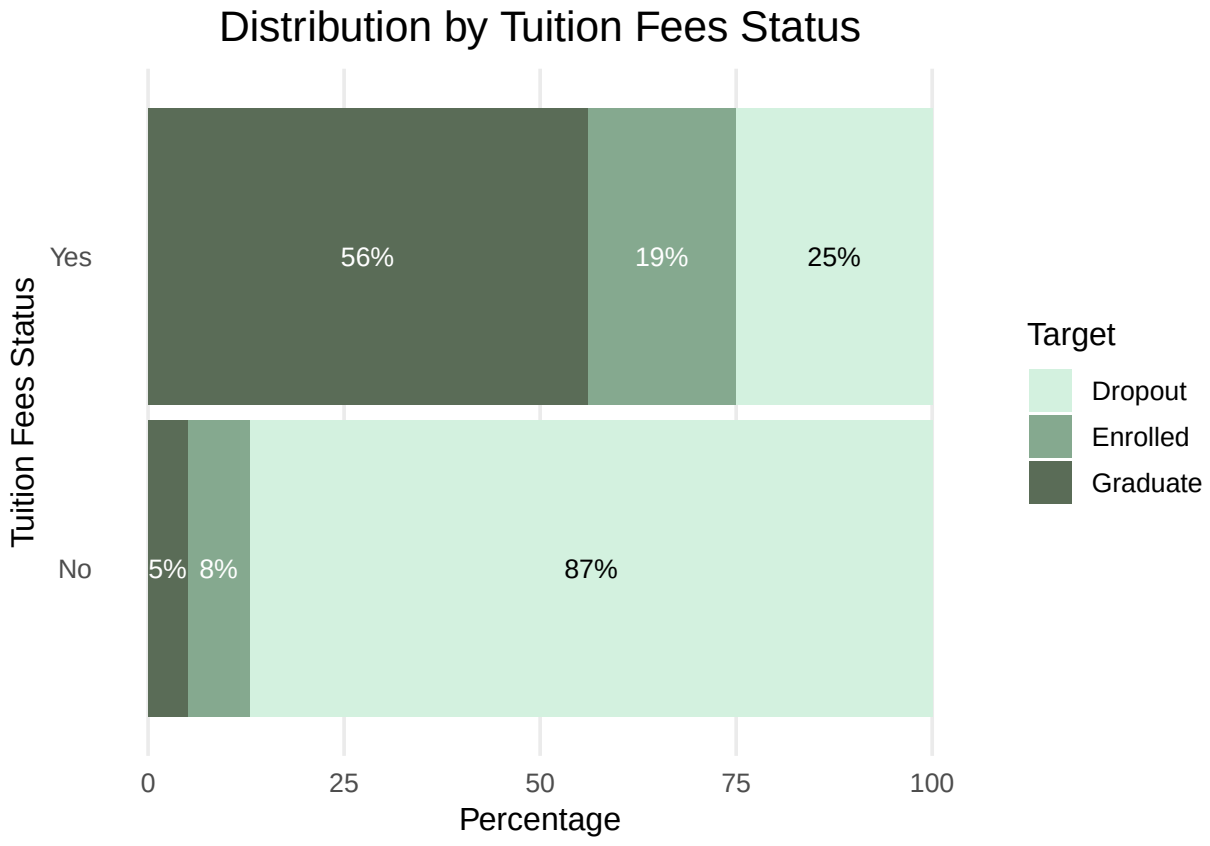



```
  labs(title = "Scholarship Holders by Target", x = "Scholarship Holder ", y = "Count") +
  theme_minimal()
```

```
## NULL
```

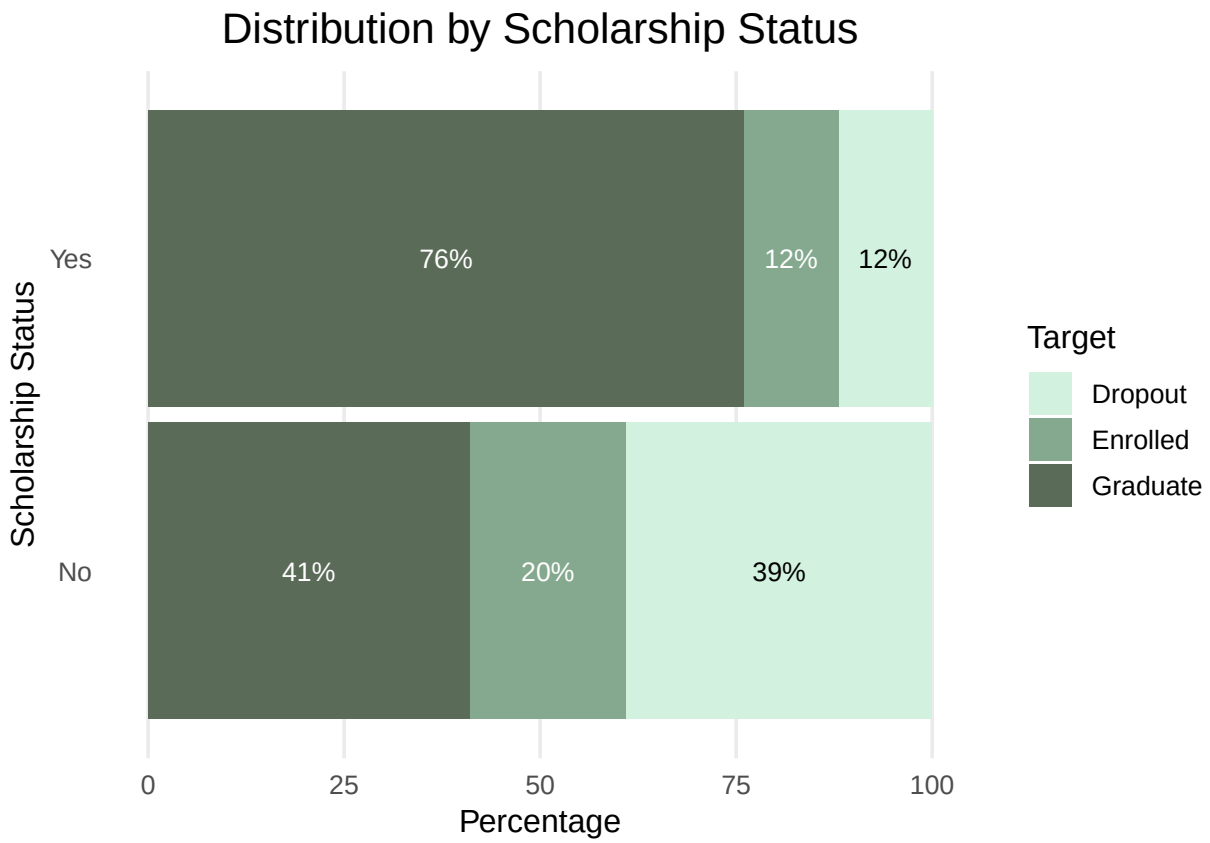
```
# Plot for Tuition Fees Up to Date
tuition_plot <- data %>%
  group_by(Tuition.fees.up.to.date, Target) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by(Tuition.fees.up.to.date) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100)) %>%
  ungroup()

ggplot(tuition_plot, aes(x = Percentage, y = Tuition.fees.up.to.date, fill = Target)) +
  geom_bar(stat = "identity", position = "stack", width = 0.95) +
  geom_text(
    aes(
      label = paste0(Percentage, "%"),
      color = ifelse(Target == "Dropout", "black", "white") # Conditional text color
    ),
    position = position_stack(vjust = 0.5), # Center the text vertically in the box
    size = 3.5
  ) +
  labs(
    title = "Distribution by Tuition Fees Status",
```

```
    x = "Percentage",
    y = "Tuition Fees Status"
  ) +
  scale_fill_manual(
    values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"),
    name = "Target"
  ) +
  scale_color_identity() + # Use specified text colors directly
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank()
  )
```



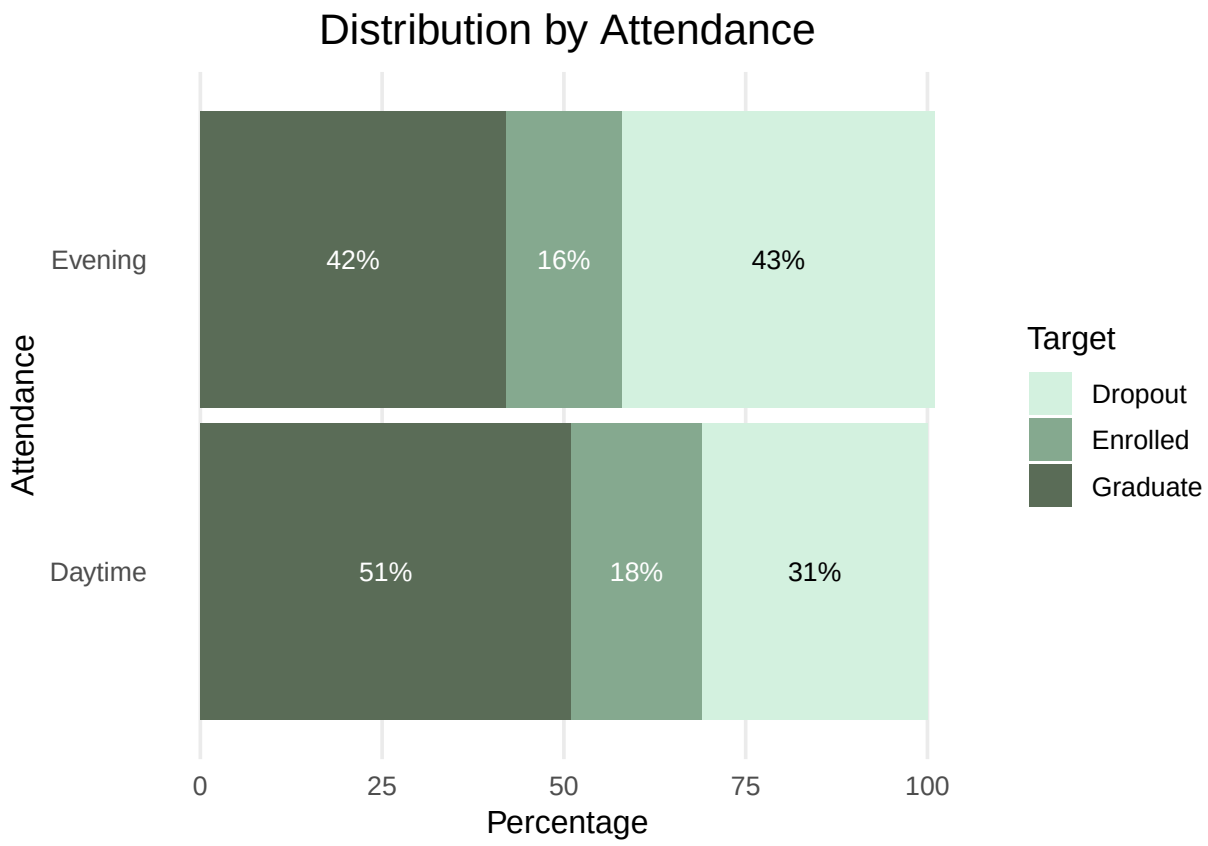

```
# Plot for Scholarship Holder
scholarship_plot <- data %>%
  group_by(Scholarship.holder, Target) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by(Scholarship.holder) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100)) %>%
```

```r
  ungroup()

ggplot(scholarship_plot, aes(x = Percentage, y = Scholarship.holder, fill = Target)) +
  geom_bar(stat = "identity", position = "stack", width = 0.95) +
  geom_text(
    aes(
      label = paste0(Percentage, "%"),
      color = ifelse(Target == "Dropout", "black", "white") # Conditional text color
    ),
    position = position_stack(vjust = 0.5), # Center the text vertically in the box
    size = 3.5
  )+
  labs(
    title = "Distribution by Scholarship Status",
    x = "Percentage",
    y = "Scholarship Status"
  ) +
 scale_fill_manual(
    values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"),
    name = "Target"
  ) +
  scale_color_identity() + # Use specified text colors directly
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank()
  )
```



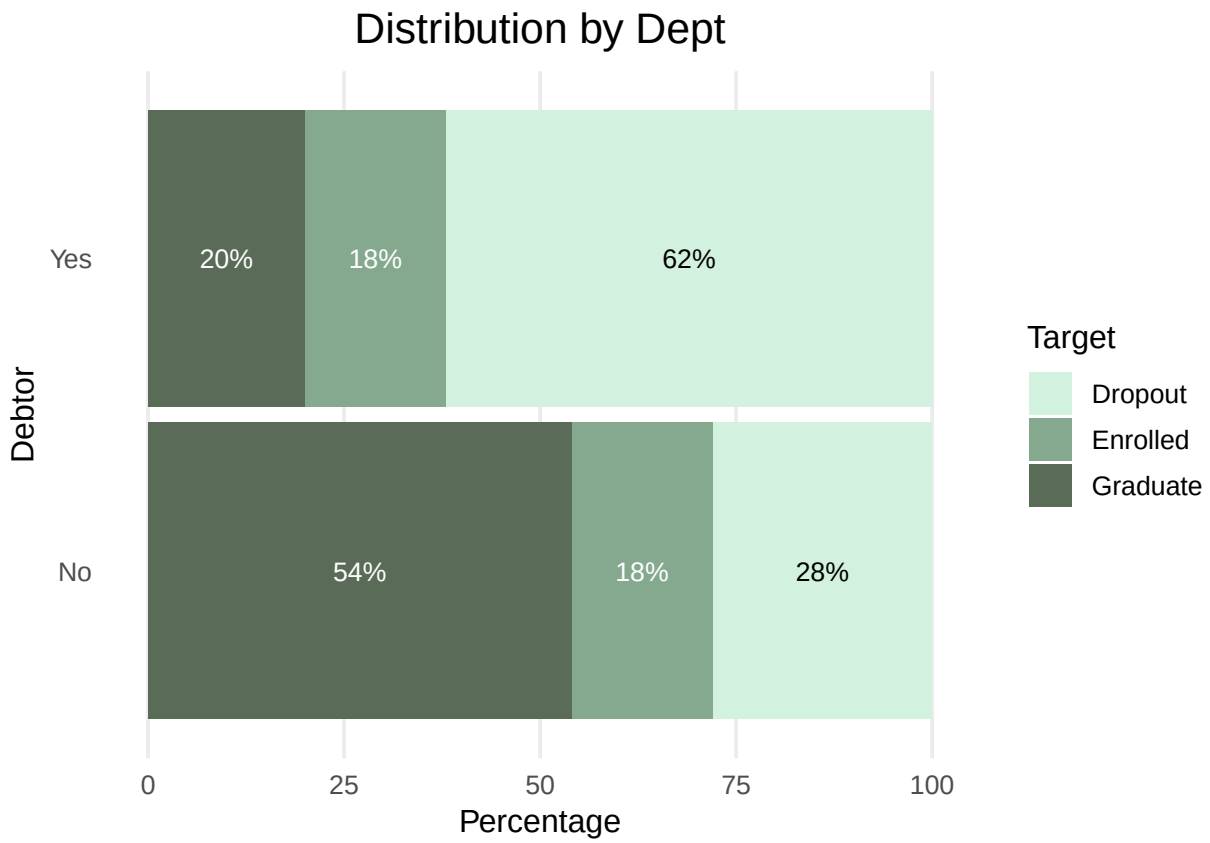

```r
# Plot for Daytime/Evening Attendance
attendance_plot <- data %>%
  group_by(Daytime.evening.attendance, Target) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by(Daytime.evening.attendance) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100)) %>%
  ungroup()

ggplot(attendance_plot, aes(x = Percentage, y = Daytime.evening.attendance, fill = Target)) +
  geom_bar(stat = "identity", position = "stack", width = 0.95) +
  geom_text(
    aes(
      label = paste0(Percentage, "%"),
      color = ifelse(Target == "Dropout", "black", "white") # Conditional text color
    ),
    position = position_stack(vjust = 0.5), # Center the text vertically in the box
    size = 3.5
  )+
  labs(
    title = "Distribution by Attendance",
    x = "Percentage",
    y = "Attendance"
  ) +
  scale_fill_manual(
    values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"),
    name = "Target"
```

```
  ) +
  scale_color_identity() + # Use specified text colors directly
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank()
  )
```



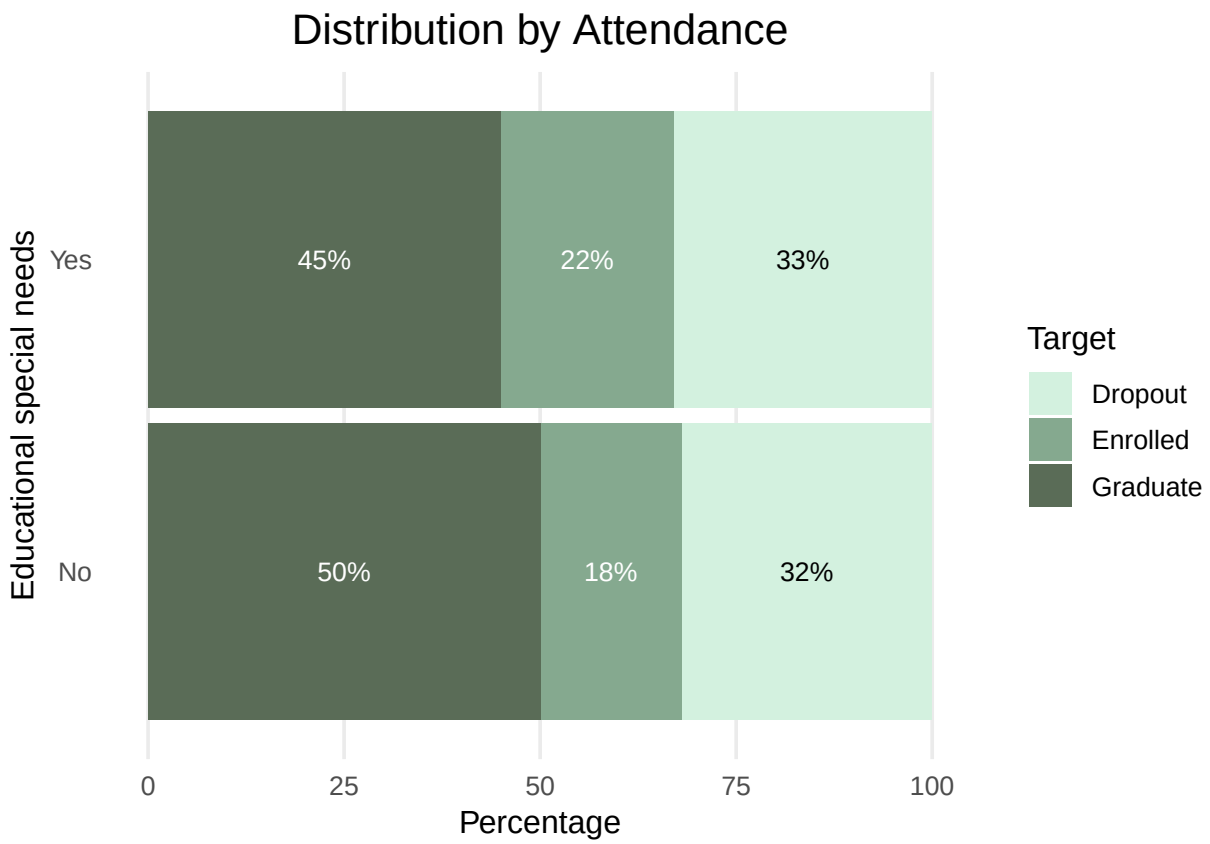

```
# Plot for Daytime/Evening Attendance
attendance_plot <- data %>%
  group_by(Debtor, Target) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by(Debtor) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100)) %>%
  ungroup()

ggplot(attendance_plot, aes(x = Percentage, y = Debtor, fill = Target)) +
  geom_bar(stat = "identity", position = "stack", width = 0.95) +
  geom_text(
    aes(
```

```
    label = paste0(Percentage, "%"),
    color = ifelse(Target == "Dropout", "black", "white") # Conditional text color
  ),
  position = position_stack(vjust = 0.5), # Center the text vertically in the box
  size = 3.5
)+
labs(
  title = "Distribution by Dept",
  x = "Percentage",
  y = "Debtor"
) +
scale_fill_manual(
  values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"),
  name = "Target"
) +
scale_color_identity() + # Use specified text colors directly
theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(hjust = 0.5, size = 16),
  axis.text = element_text(size = 10),
  axis.title = element_text(size = 12),
  legend.text = element_text(size = 10),
  legend.title = element_text(size = 12),
  panel.grid.minor = element_blank(),
  panel.grid.major.y = element_blank()
)
```



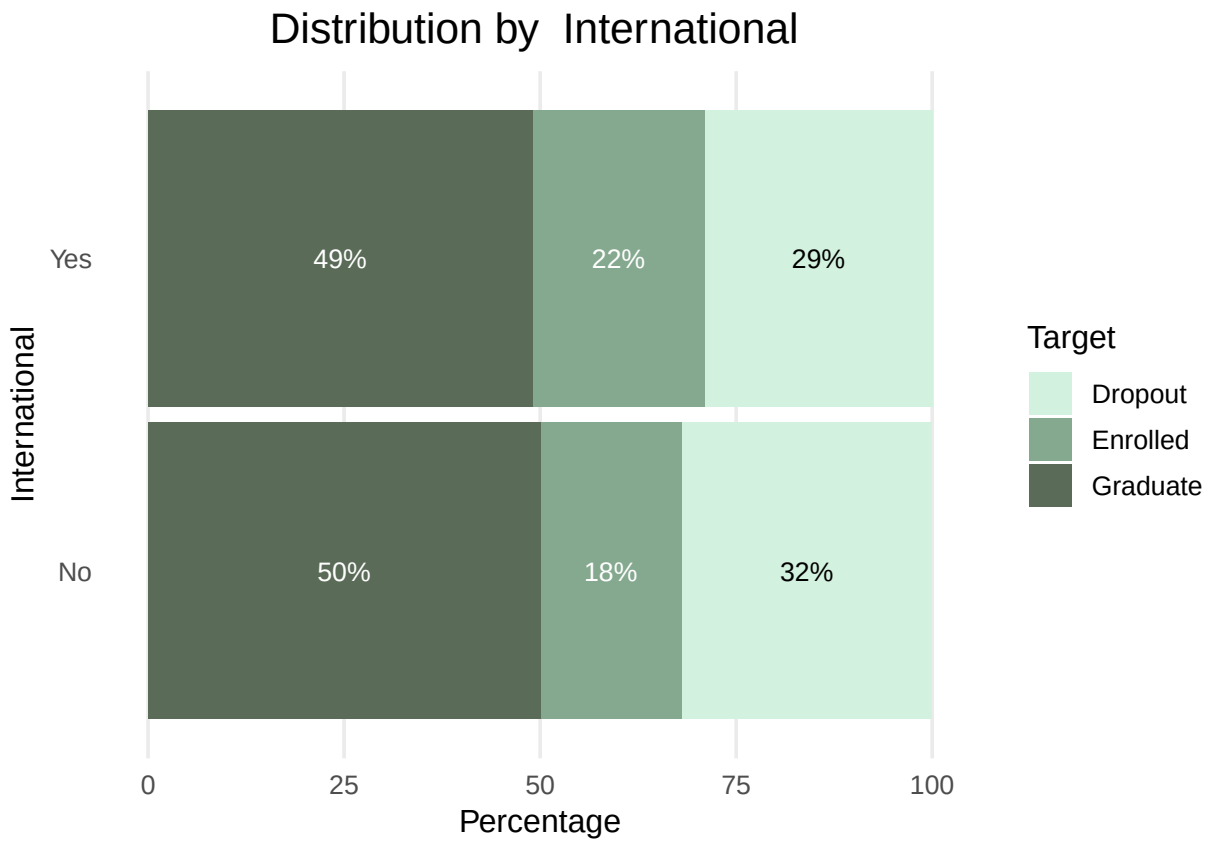

```r
# Plot for Daytime/Evening Attendance
attendance_plot <- data %>%
  group_by(Educational.special.needs, Target) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by(Educational.special.needs) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100)) %>%
  ungroup()

ggplot(attendance_plot, aes(x = Percentage, y = Educational.special.needs, fill = Target)) +
  geom_bar(stat = "identity", position = "stack", width = 0.95) +
  geom_text(
    aes(
      label = paste0(Percentage, "%"),
      color = ifelse(Target == "Dropout", "black", "white") # Conditional text color
    ),
    position = position_stack(vjust = 0.5), # Center the text vertically in the box
    size = 3.5
  )+
  labs(
    title = "Distribution by Attendance",
    x = "Percentage",
    y = "Educational special needs"
  ) +
  scale_fill_manual(
    values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"),
    name = "Target"
```

```
  ) +
  scale_color_identity() + # Use specified text colors directly
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12),
    panel.grid.minor = element_blank(),
    panel.grid.major.y = element_blank()
  )
```



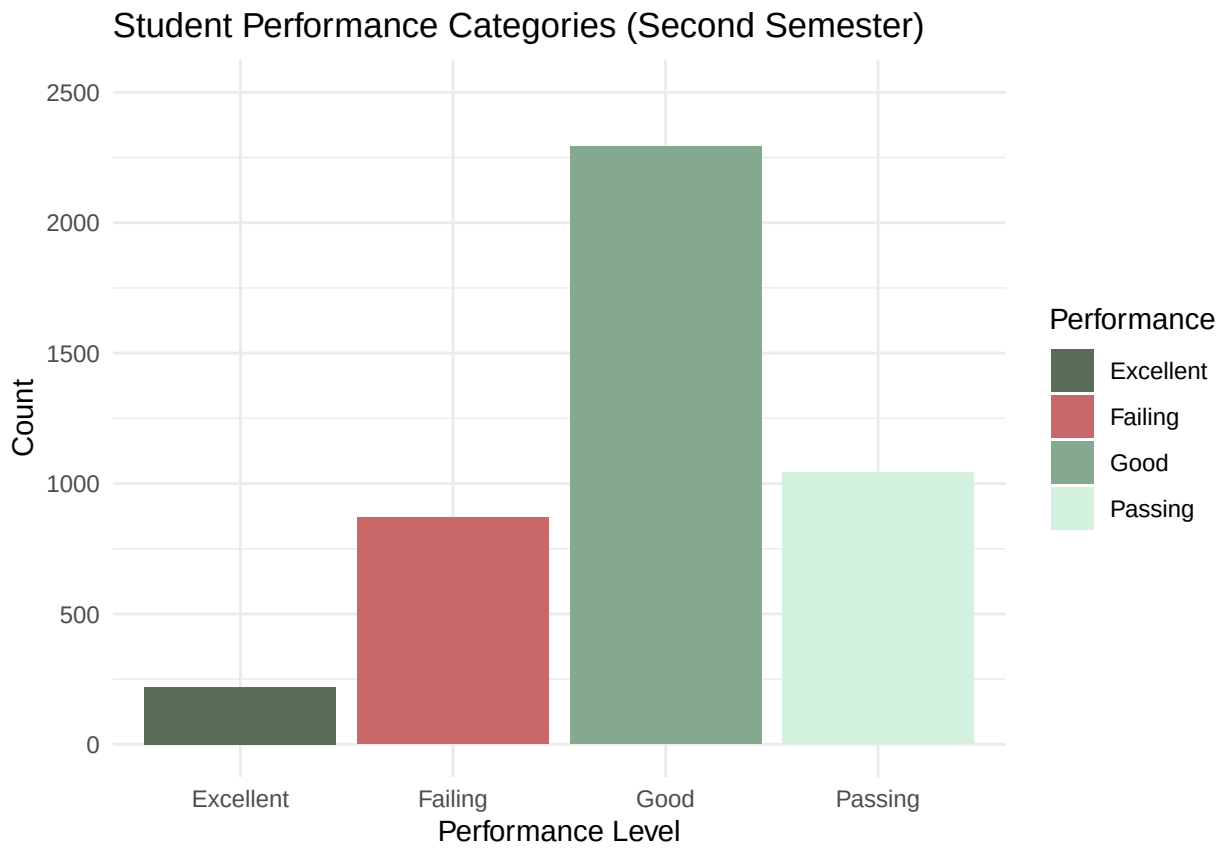

```
# Plot for Daytime/Evening Attendance
attendance_plot <- data %>%
  group_by( International , Target) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  group_by( International ) %>%
  mutate(Percentage = round((Count / sum(Count)) * 100)) %>%
  ungroup()

ggplot(attendance_plot, aes(x = Percentage, y =  International , fill = Target)) +
  geom_bar(stat = "identity", position = "stack", width = 0.95) +
  geom_text(
    aes(
```

```
    label = paste0(Percentage, "%"),
    color = ifelse(Target == "Dropout", "black", "white") # Conditional text color
  ),
  position = position_stack(vjust = 0.5), # Center the text vertically in the box
  size = 3.5
)+
labs(
  title = "Distribution by  International ",
  x = "Percentage",
  y = " International "
) +
scale_fill_manual(
  values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F"),
  name = "Target"
) +
scale_color_identity() + # Use specified text colors directly
theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(hjust = 0.5, size = 16),
  axis.text = element_text(size = 10),
  axis.title = element_text(size = 12),
  legend.text = element_text(size = 10),
  legend.title = element_text(size = 12),
  panel.grid.minor = element_blank(),
  panel.grid.major.y = element_blank()
)
```




```r
library(ggplot2)

# Define custom colors
custom_colors <- c(
  "Failing" = "#C96868",  # Red
  "Passing" = "#D3F1DF",  # Yellow
  "Good" = "#85A98F",     # Green
  "Excellent" = "#5A6C57" # Blue
)

# Plot for First Semester Performance
ggplot(data, aes(x = First.Sem.Performance, fill =First.Sem.Performance)) +
  geom_bar() +
  scale_fill_manual(values = custom_colors) +  # Use custom colors
  labs(
    title = "Student Performance Categories (First Semester)",
    x = "Performance Level",
    y = "Count",
    fill = "Performance"
  ) +
  ylim(0, 2500) +  # Set a limit for the y-axis
  theme_minimal()
```

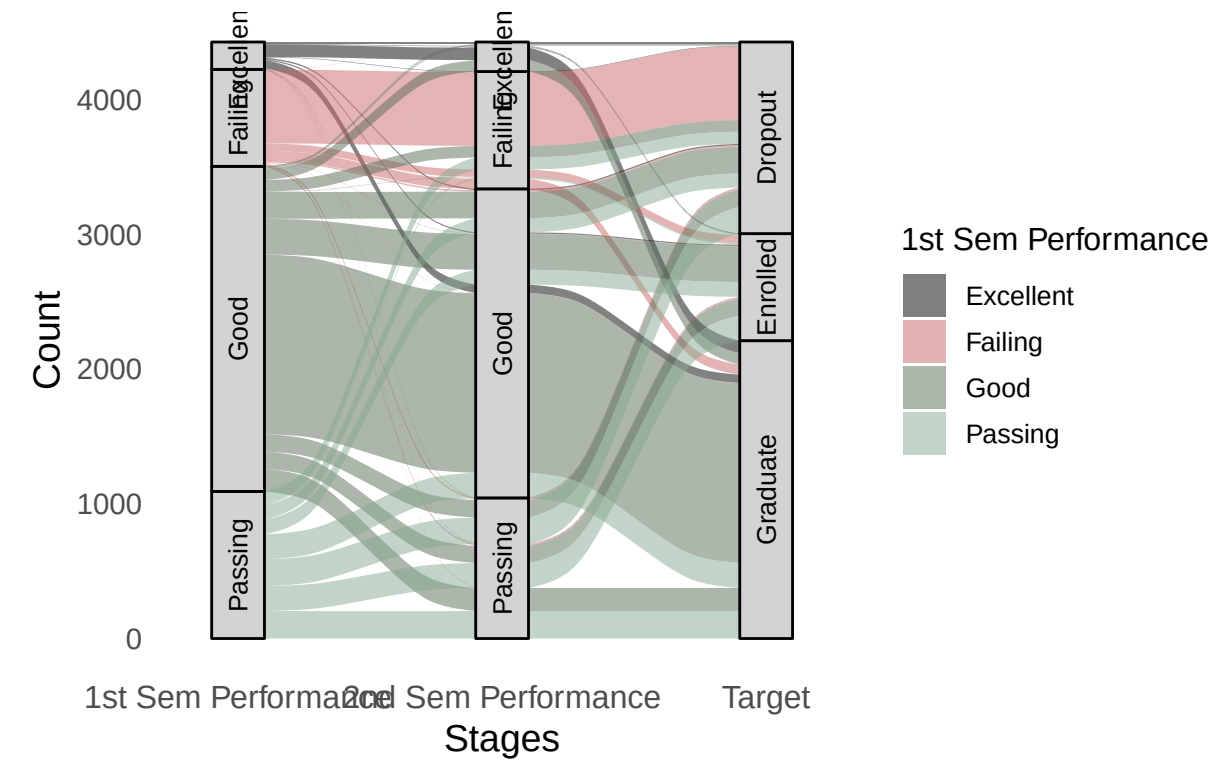

```r
# Plot for Second Semester Performance
ggplot(data, aes(x = Second.Sem.Performance, fill = Second.Sem.Performance)) +
  geom_bar() +
  scale_fill_manual(values = custom_colors) +  # Use custom colors
  labs(
    title = "Student Performance Categories (Second Semester)",
    x = "Performance Level",
    y = "Count",
    fill = "Performance"
  ) +
  ylim(0, 2500) +  # Set a limit for the y-axis
  theme_minimal()
```




```r
parallel_data <- data %>%
  select(First.Sem.Performance, Second.Sem.Performance, Target) %>%
  na.omit() %>%
  group_by(First.Sem.Performance, Second.Sem.Performance, Target) %>%
  summarise(Count = n(), .groups = "drop")

# Custom colors for First_Sem_Performance categories
custom_colors <- c(
  "Failing" = "#C96868",  # Red
  "Passing" = "#85A98F",  # Yellow
  "Good" ="#5A6C57",     # Green
  "Excellent" = "black"
)

# Ensure the data is properly prepared
parallel_data <- data %>%
  select(First.Sem.Performance, Second.Sem.Performance, Target) %>%
  na.omit() %>%
  group_by(First.Sem.Performance, Second.Sem.Performance, Target) %>%
  summarise(Count = n(), .groups = "drop")

# Create the parallel set plot
parallel_plot <- ggplot(parallel_data, aes(
  axis1 = First.Sem.Performance,
  axis2 = Second.Sem.Performance,
  axis3 = Target,
```

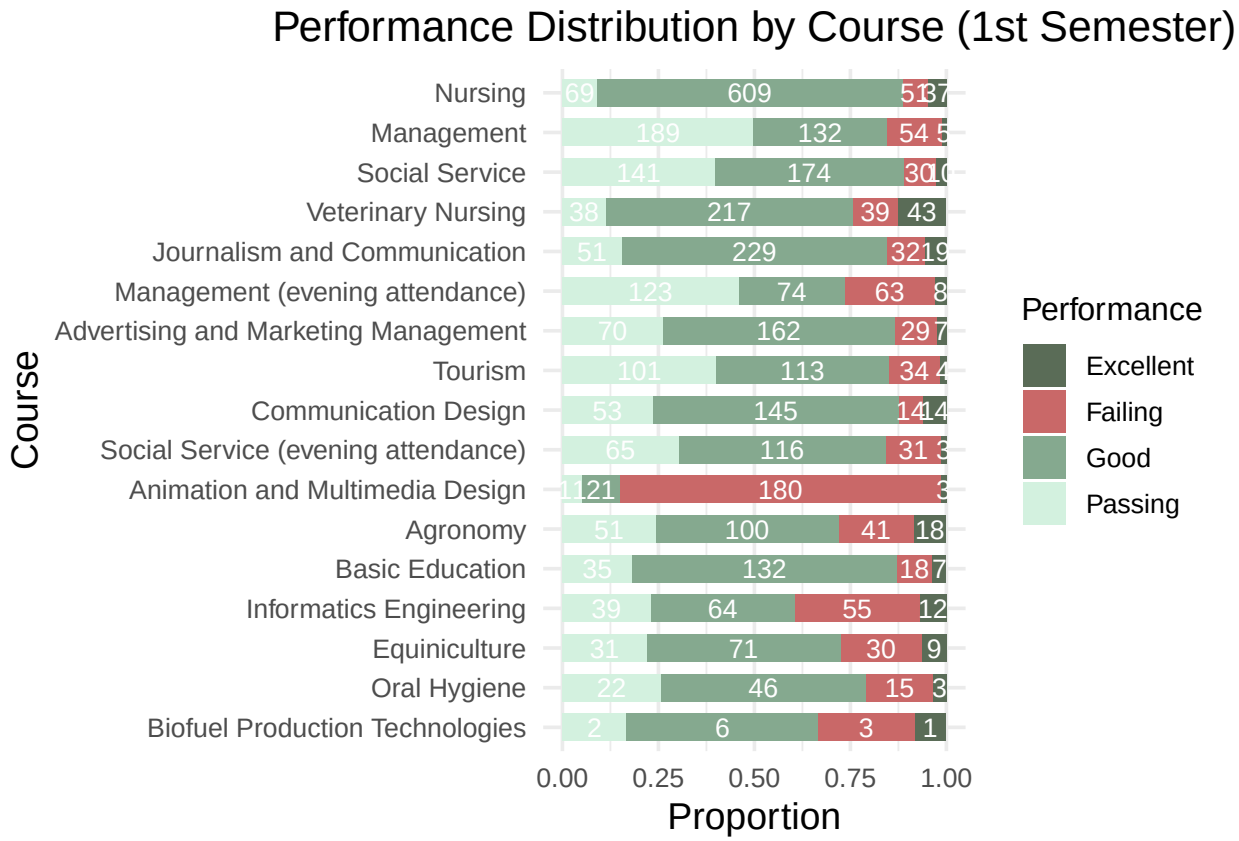
```
  y = Count  # Map count to y for proper sizing
)) +
  geom_alluvium(aes(fill = First.Sem.Performance), width = 1/5) +
  geom_stratum(width = 1/5, fill = "lightgray", color = "black") +
  geom_text(
    stat = "stratum",
    aes(label = after_stat(stratum)),
    angle = 90,  # Make text vertical
    size = 3.5,  # Adjust text size
    color = "black",  # Ensure text is readable
    hjust = 0.5,  # Center-align horizontally within the box
    vjust = 0.5   # Center-align vertically within the box
  ) +
  scale_x_discrete(
    limits = c("1st Sem Performance", "2nd Sem Performance", "Target"),
    expand = c(0.1, 0.1)  # Add space between value boxes
  ) +
  scale_fill_manual(values = custom_colors) +
  labs(
    title = "Parallel Set Chart: Performance Flow for Students",
    x = "Stages",
    y = "Count",
    fill = "1st Sem Performance"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18),
    axis.text.x = element_text(size = 12),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12),
    panel.grid = element_blank()  # Remove grid for cleaner look
  ) +
  theme(plot.margin = margin(t = 10, r = 10, b = 10, l = 10))

# Display the plot
parallel_plot
```

```
## Warning in to_lodes_form(data = data, axes = axis_ind, discern =
## params$discern): Some strata appear at multiple axes.
## Warning in to_lodes_form(data = data, axes = axis_ind, discern =
## params$discern): Some strata appear at multiple axes.
## Warning in to_lodes_form(data = data, axes = axis_ind, discern =
## params$discern): Some strata appear at multiple axes.
```



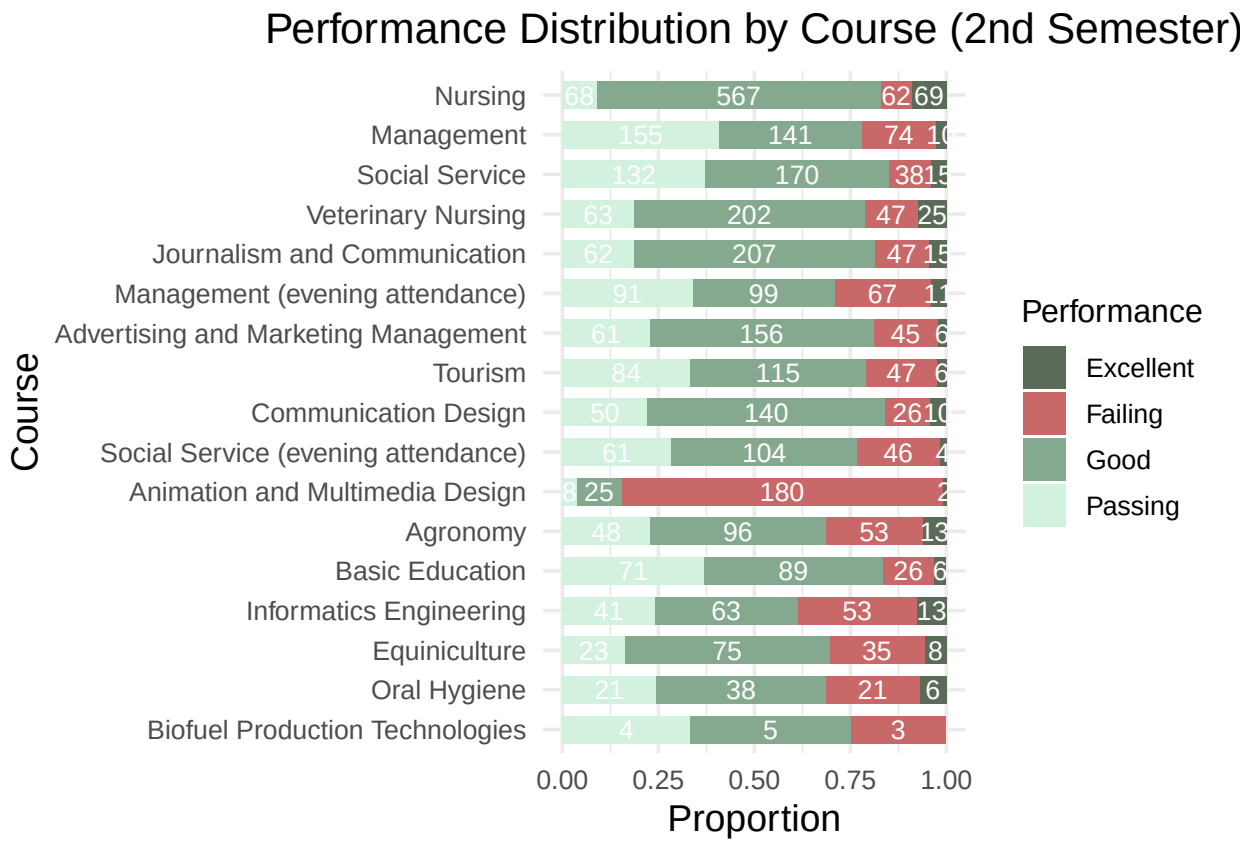

```r
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Aggregate data by Course and First Semester Performance
first.Sem.performance <- data %>%
  group_by(Course, First.Sem.Performance) %>%
  summarise(Count = n(), .groups = 'drop')

# Reorder courses based on total count for better readability
first.Sem.performance <- first.Sem.performance %>%
  group_by(Course) %>%
  mutate(Total = sum(Count)) %>%
  ungroup() %>%
  mutate(Course = reorder(Course, Total))

# Plot for First Semester
ggplot(first.Sem.performance, aes(x = Count, y = Course, fill = First.Sem.Performance)) +
  geom_bar(stat = "identity", position = "fill", width = 0.7) +  # Adjust position to "fill" for full-w
  geom_text(
    aes(label = Count),
    position = position_fill(vjust = 0.5),  # Place text in the center of the bars
    color = "white",
    size = 3.5
  ) +
  labs(
```

```
    title = "Performance Distribution by Course (1st Semester)",
    x = "Proportion",
    y = "Course",
    fill = "Performance"
  ) +
  scale_fill_manual(
    values = c("Failing" = "#C96868",  # Red
  "Passing" = "#D3F1DF",  # Yellow
  "Good" = "#85A98F",     # Green
  "Excellent" = "#5A6C57" # Blue
)
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12)
  )
```



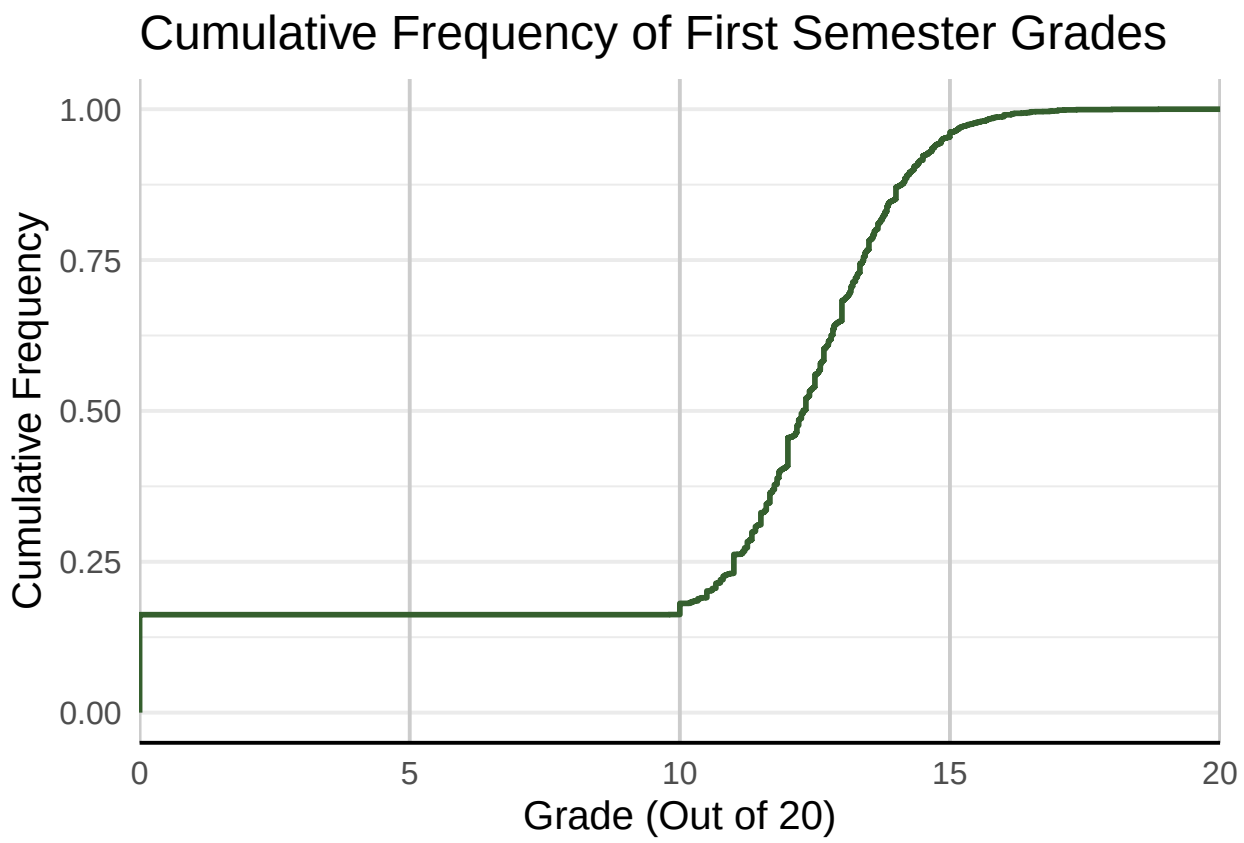

```
# Aggregate data by Course and Second Semester Performance
second.Sem.performance <- data %>%
  group_by(Course, Second.Sem.Performance) %>%
  summarise(Count = n(), .groups = 'drop')
```

```r
# Reorder courses based on total count for better readability
second.Sem.performance <- second.Sem.performance %>%
  group_by(Course) %>%
  mutate(Total = sum(Count)) %>%
  ungroup() %>%
  mutate(Course = reorder(Course, Total))

# Plot for Second Semester
ggplot(second.Sem.performance, aes(x = Count, y = Course, fill = Second.Sem.Performance)) +
  geom_bar(stat = "identity", position = "fill", width = 0.7) +  # Adjust position to "fill" for full-w
  geom_text(
    aes(label = Count),
    position = position_fill(vjust = 0.5),  # Place text in the center of the bars
    color = "white",
    size = 3.5
  ) +
  labs(
    title = "Performance Distribution by Course (2nd Semester)",
    x = "Proportion",
    y = "Course",
    fill = "Performance"
  ) +
  scale_fill_manual(
    values = c("Failing" = "#C96868",  # Red
  "Passing" = "#D3F1DF",  # Yellow
  "Good" = "#85A98F",     # Green
  "Excellent" = "#5A6C57" # Blue
)
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12)
  )
```

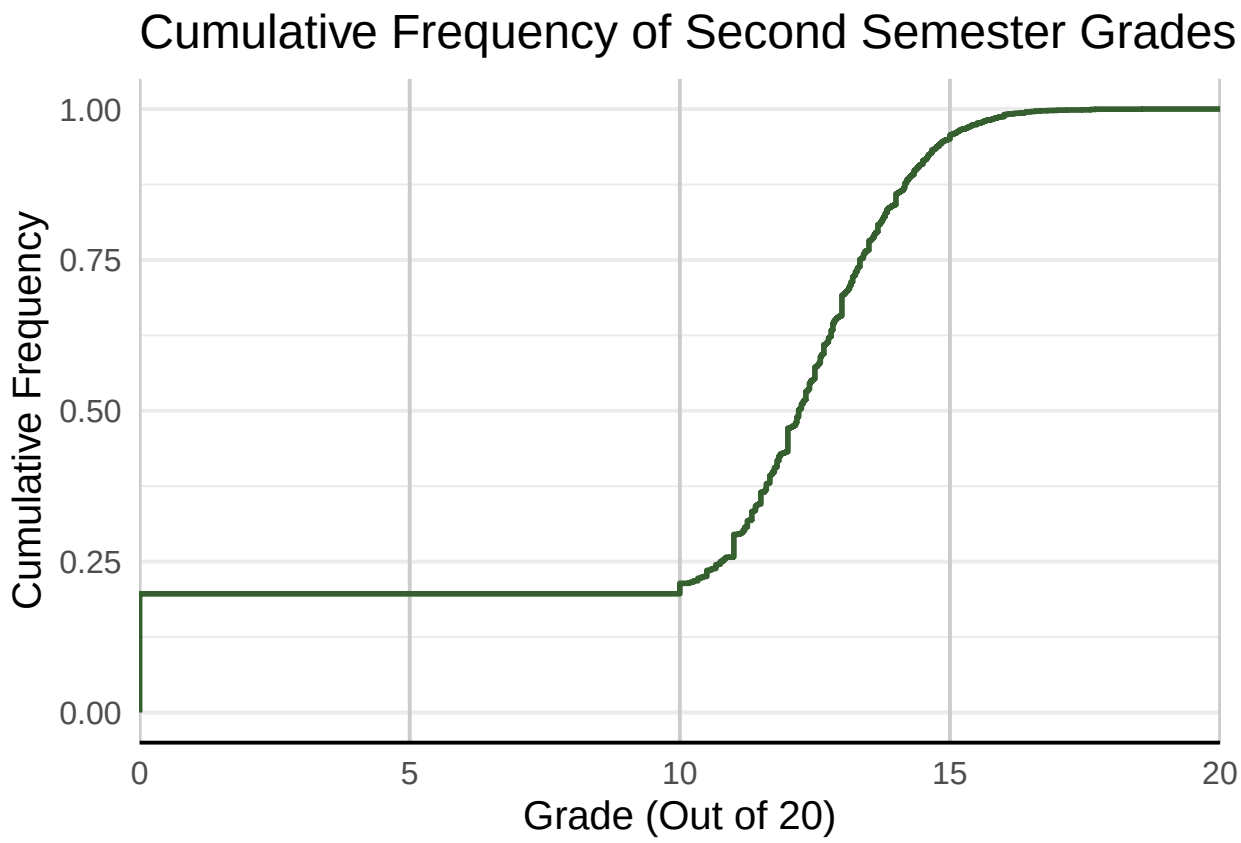



```r
library(ggplot2)
library(dplyr)

# Ensure grades are numeric and filter for valid grades
data <- data %>%
  mutate(
    `Curricular.units.1st.sem..grade.` = as.numeric(`Curricular.units.1st.sem..grade.`),
    `Curricular.units.2nd.sem..grade.` = as.numeric(`Curricular.units.2nd.sem..grade.`)
  ) %>%
  filter(`Curricular.units.1st.sem..grade.` >= 0 & `Curricular.units.2nd.sem..grade.` >= 0)

# Create cumulative frequency plot for first semester grades
ggplot(data, aes(x = `Curricular.units.1st.sem..grade.`)) +
  stat_ecdf(geom = "step", color = "#355F2E", size = 1) +
  labs(
    title = "Cumulative Frequency of First Semester Grades",
    x = "Grade (Out of 20)",
    y = "Cumulative Frequency"
  ) +
  scale_x_continuous(limits = c(0, 20), expand = c(0, 0)) + # Remove padding before zero
  theme_minimal(base_size = 15) +
  theme(
    panel.grid.minor.x = element_blank(), # Remove minor gridlines on x-axis
    panel.grid.major.x = element_line(color = "gray80"), # Optional: Customize major gridlines
    axis.line.x = element_line(color = "black") # Ensure the x-axis line is emphasized
  )
```

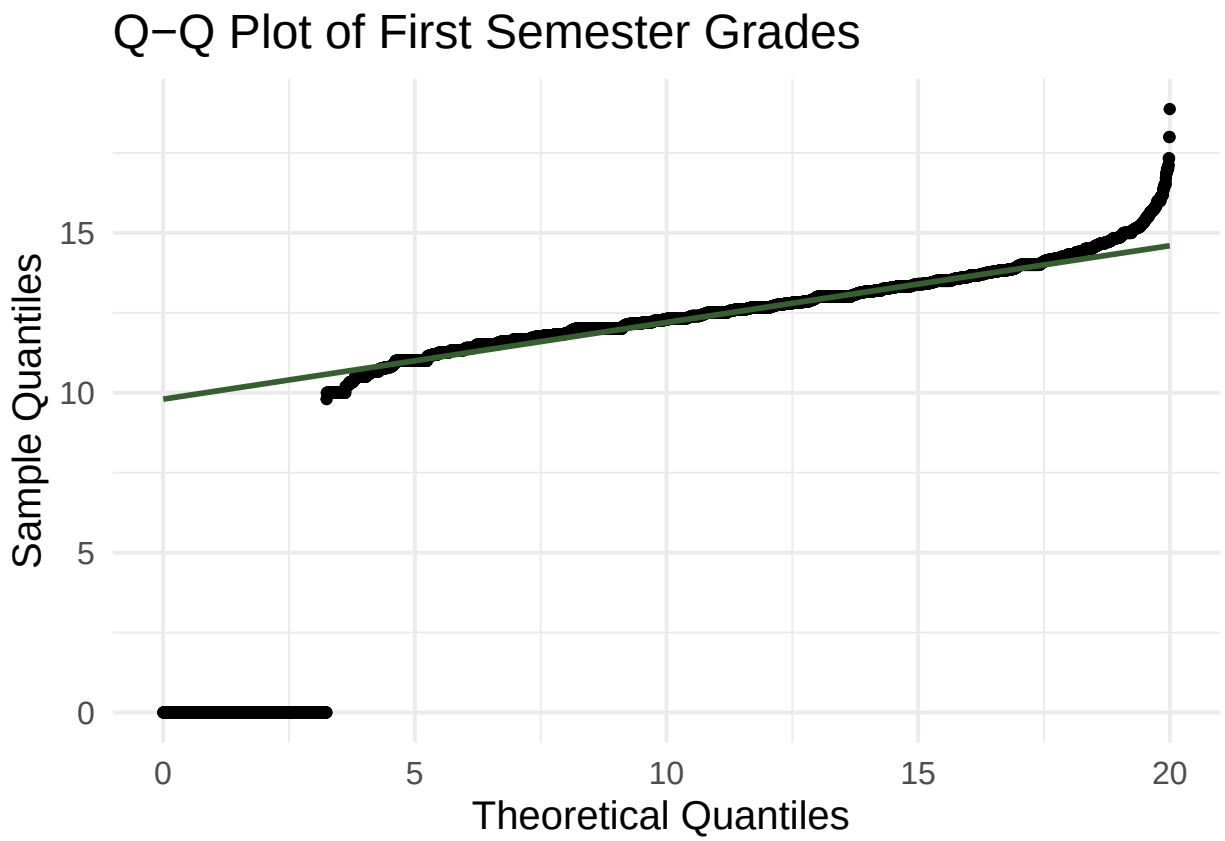
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



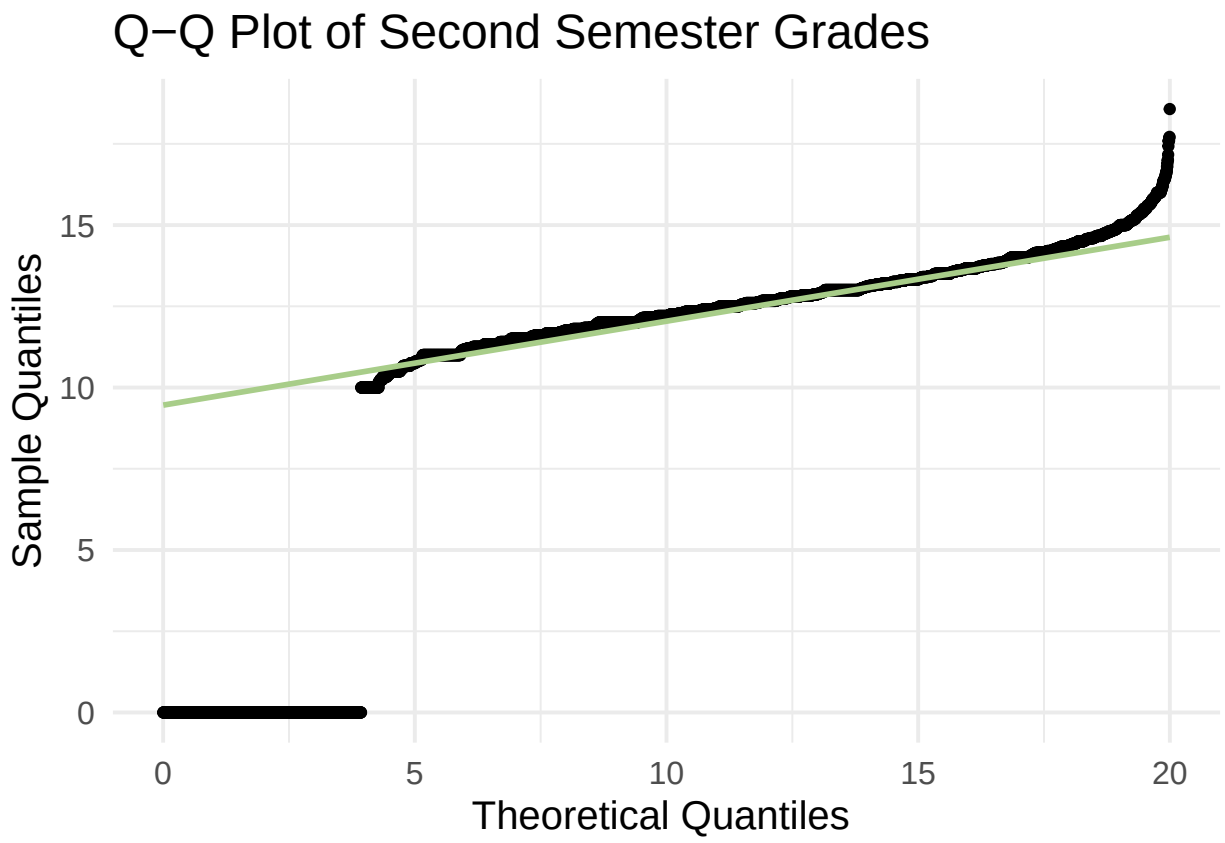

```r
# Create cumulative frequency plot for second semester grades
ggplot(data, aes(x = `Curricular.units.2nd.sem..grade.`)) +
  stat_ecdf(geom = "step", color = "#355F2E", size = 1) +
  labs(
    title = "Cumulative Frequency of Second Semester Grades",
    x = "Grade (Out of 20)",
    y = "Cumulative Frequency"
  ) +
  scale_x_continuous(limits = c(0, 20), expand = c(0, 0)) + # Remove padding before zero
  theme_minimal(base_size = 15) +
  theme(
    panel.grid.minor.x = element_blank(), # Remove minor gridlines on x-axis
    panel.grid.major.x = element_line(color = "gray80"), # Optional: Customize major gridlines
    axis.line.x = element_line(color = "black") # Ensure the x-axis line is emphasized
  )
```



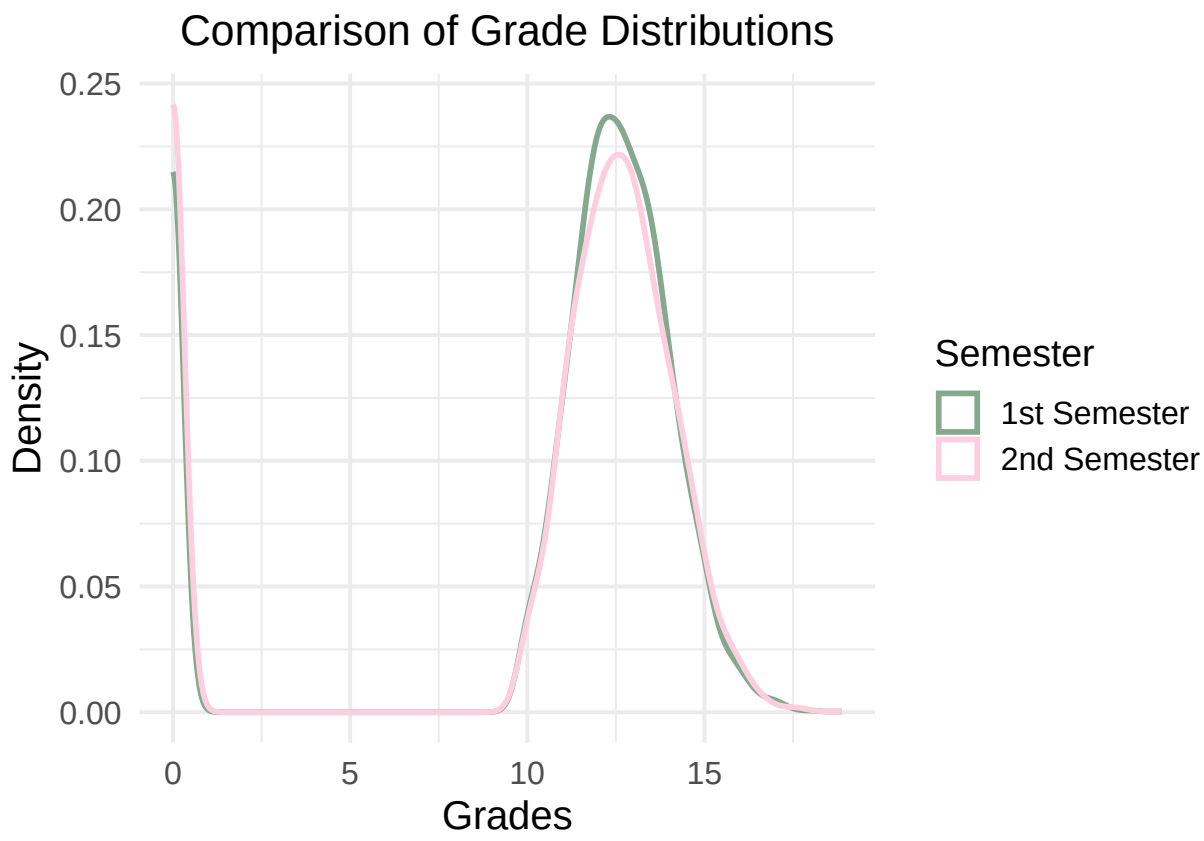

```r
library(ggplot2)
library(dplyr)

# Ensure grades are numeric
data <- data %>%
  mutate(
    `Curricular.units.1st.sem..grade.` = as.numeric(`Curricular.units.1st.sem..grade.`),
    `Curricular.units.2nd.sem..grade.` = as.numeric(`Curricular.units.2nd.sem..grade.`)
  )

# Q-Q plot for the first semester grades (against a uniform distribution)
ggplot(data, aes(sample = `Curricular.units.1st.sem..grade.`)) +
  stat_qq(distribution = function(p) qunif(p, min = 0, max = 20)) +
  stat_qq_line(distribution = function(p) qunif(p, min = 0, max = 20), color = "#355F2E", size = 1) +
  labs(
    title = "Q-Q Plot of First Semester Grades ",
    x = "Theoretical Quantiles ",
    y = "Sample Quantiles"
  ) +
  theme_minimal(base_size = 15)+
  theme(
    legend.position = "center"
  )
```




```r
# Q-Q plot for the second semester grades
ggplot(data, aes(sample = `Curricular.units.2nd.sem..grade.`)) +
  stat_qq(distribution = function(p) qunif(p, min = 0, max = 20)) +
  stat_qq_line(distribution = function(p) qunif(p, min = 0, max = 20), color = "#A8CD89", size = 1) +
  labs(
    title = "Q-Q Plot of Second Semester Grades ",
    x = "Theoretical Quantiles ",
    y = "Sample Quantiles"
  ) +
  theme_minimal(base_size = 15)
```

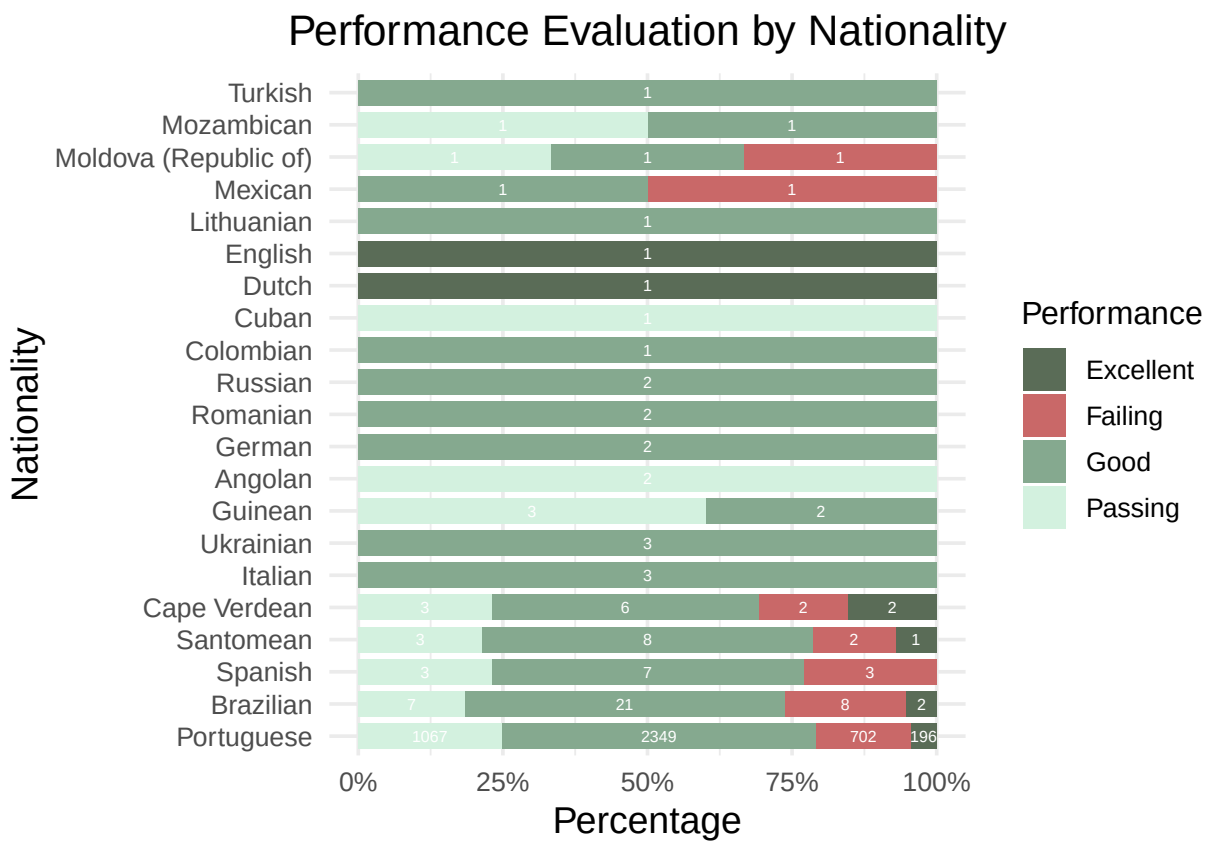



```r
# Overlay density plots for 1st and 2nd semesters with custom colors
ggplot() +
  geom_density(data = data, aes(x = Curricular.units.1st.sem..grade., color = "1st Semester"), size = 1
  geom_density(data = data, aes(x = Curricular.units.2nd.sem..grade., color = "2nd Semester"), size = 1
  scale_color_manual(
    values = c("1st Semester" = "#85A98F",  # Custom color for 1st Semester (blue)
               "2nd Semester" = "#FDCEDF")  # Custom color for 2nd Semester (orange)
  ) +
  labs(
    title = "Comparison of Grade Distributions",
    x = "Grades",
    y = "Density",
    color = "Semester"
  ) +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text = element_text(size = 12),
    legend.text = element_text(size = 12),
    legend.title = element_text(size = 14)
  )
```




```r
library(ggplot2)
library(dplyr)

# Aggregate data by Nationality and Performance
performance_nationality <- data %>%
  group_by(Nacionality, First.Sem.Performance) %>%
  summarise(Count = n(), .groups = "drop") %>%
  mutate(Percentage = Count / sum(Count) * 100)

# Plotting the data
ggplot(performance_nationality, aes(y = reorder(Nacionality, -Count), x = Percentage, fill = First.Sem.
  geom_bar(stat = "identity", position = "fill", width = 0.8) +
  geom_text(
    aes(label = Count),
    position = position_fill(vjust = 0.5),
    size = 2,
    color = "white"
  ) +
  scale_x_continuous(labels = scales::percent) +
  scale_fill_manual(
    values =c("Failing" = "#C96868",  # Red
  "Passing" = "#D3F1DF",  # Yellow
  "Good" = "#85A98F",     # Green
  "Excellent" = "#5A6C57"
  )) +
  labs(
```

```
    title = "Performance Evaluation by Nationality",
    x = "Percentage",
    y = "Nationality",
    fill = "Performance"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.y = element_text(size = 10),
    axis.text.x = element_text(size = 10),
    plot.title = element_text(hjust = 0.5, size = 16),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12)
  )
```

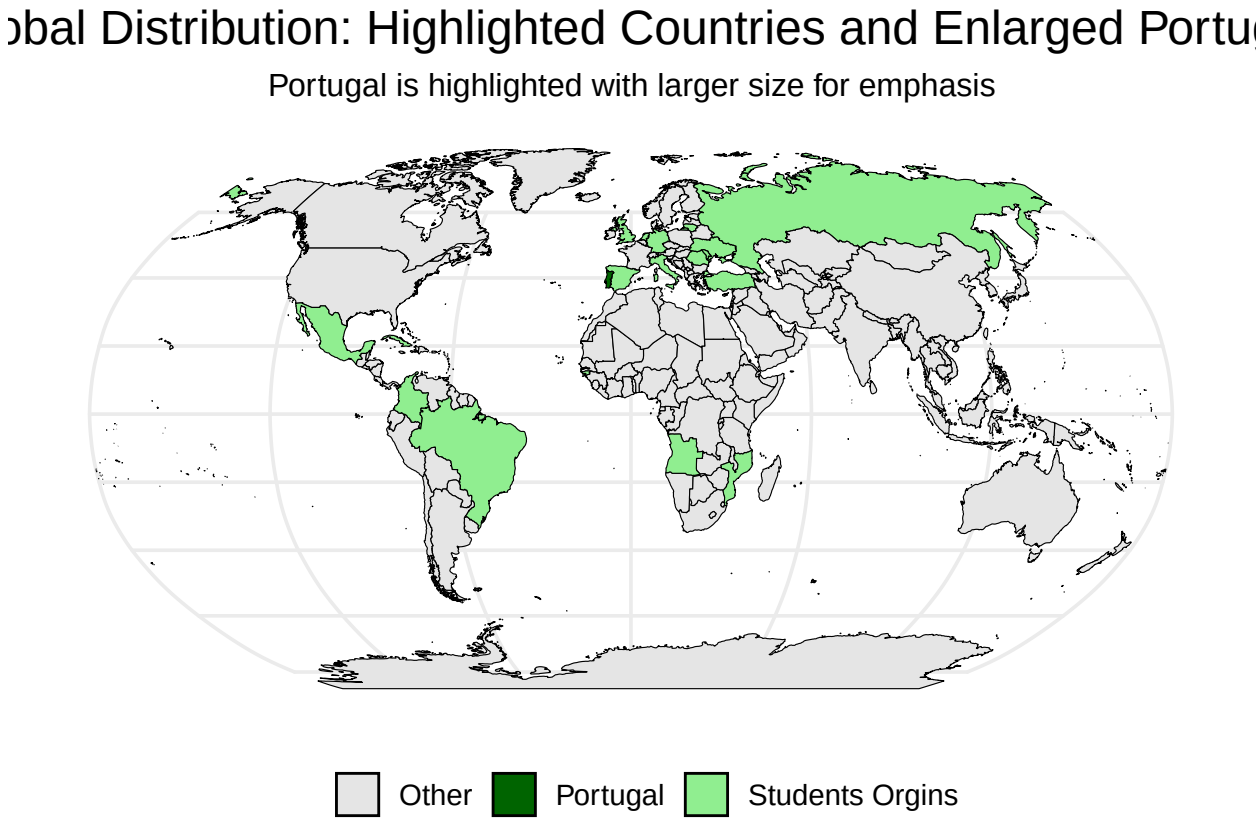



```
# Install and load necessary libraries
if (!requireNamespace("rnaturalearth", quietly = TRUE)) install.packages("rnaturalearth")
if (!requireNamespace("sf", quietly = TRUE)) install.packages("sf")
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2")
if (!requireNamespace("cartogram", quietly = TRUE)) install.packages("cartogram")

library(rnaturalearth)
library(sf)
library(ggplot2)
library(dplyr)
library(cartogram)
```

```
##
## Attaching package: 'cartogram'

## The following object is masked from 'package:terra':
##
##     cartogram
```

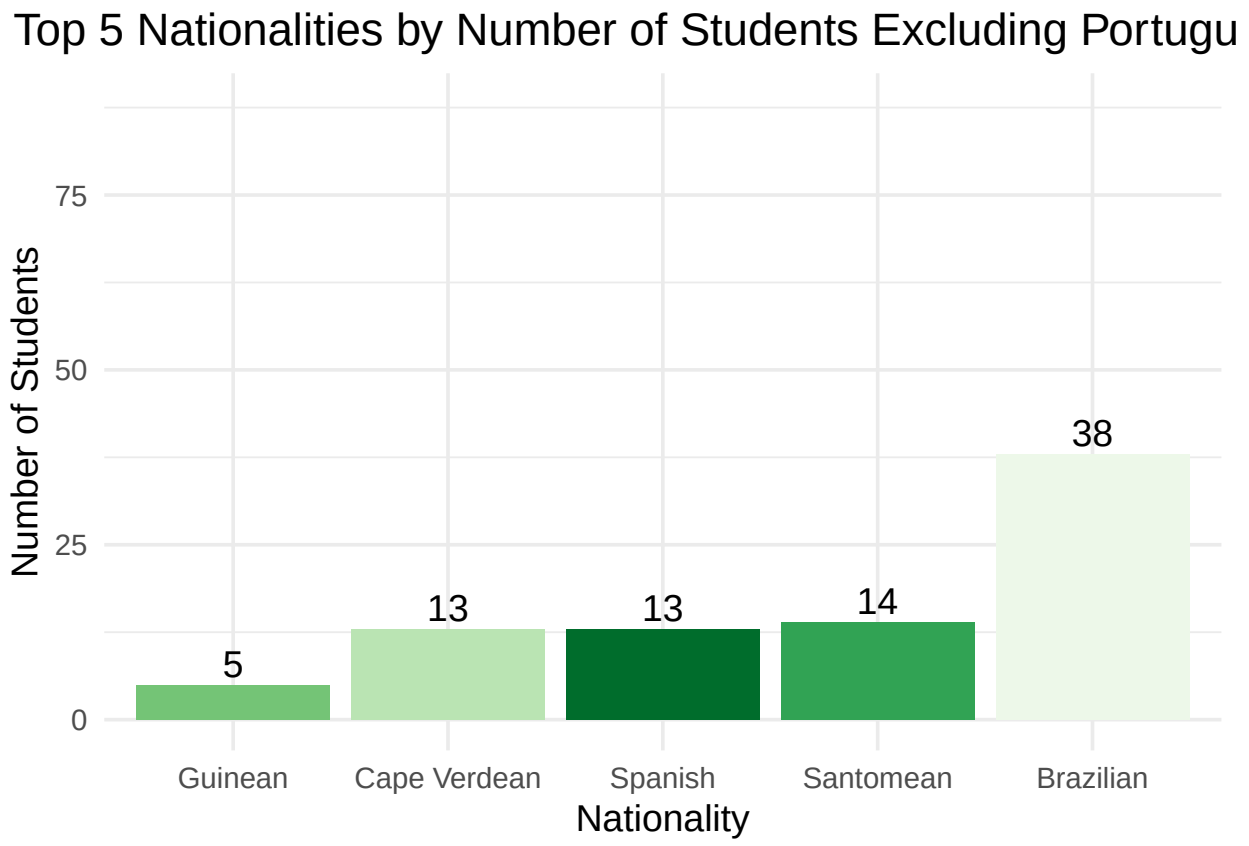
```r
# Load world map
world <- ne_countries(scale = "medium", returnclass = "sf")

# Map nationality codes to country names
nationality_mapping <- data.frame(
  Nacionality = c(
    "Portuguese", "German", "Spanish", "Italian", "Dutch", "English", "Lithuanian",
    "Angolan", "Cape Verdean", "Guinean", "Mozambican", "Santomean", "Turkish",
    "Brazilian", "Romanian", "Moldova (Republic of)", "Mexican", "Ukrainian",
    "Russian", "Cuban", "Colombian"
  ),
  Country = c(
    "Portugal", "Germany", "Spain", "Italy", "Netherlands", "United Kingdom", "Lithuania",
    "Angola", "Cabo Verde", "Guinea-Bissau", "Mozambique", "Sao Tome and Principe", "Turkey",
    "Brazil", "Romania", "Moldova", "Mexico", "Ukraine",
    "Russia", "Cuba", "Colombia"
  )
)

# Aggregate data by nationality
map_data <- data %>%
  group_by(Nacionality) %>%
  summarise(Count = n(), .groups = "drop") %>%
  left_join(nationality_mapping, by = "Nacionality")

# Join with world map data
world_data <- left_join(world, map_data, by = c("name" = "Country"))

# Highlight Portugal and other countries
world_data <- world_data %>%
  mutate(
    Highlight = case_when(
      name == "Portugal" ~ "Portugal",
      !is.na(Count) ~ "Students Orgins",
      TRUE ~ "Other"
    )
  )

# Transform the projection to Robinson
world_data_proj <- st_transform(world_data, crs = "+proj=robin")

# Plot the map
ggplot(world_data_proj) +
  geom_sf(aes(fill = Highlight), color = "black", size = 0.2) +  # Add borders
  scale_fill_manual(
    values = c(
      "Portugal" = "#006400",  # Dark green for Portugal
```

```
      "Students Orgins" = "#90EE90",  # Light green for other countries in dataset
      "Other" = "grey90"  # Grey for other countries
    ),
    name = " "
  ) +
  coord_sf(crs = st_crs("+proj=robin")) +  # Robinson projection
  labs(
    title = "Global Distribution: Highlighted Countries and Enlarged Portugal",
    subtitle = "Portugal is highlighted with larger size for emphasis"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18),
    plot.subtitle = element_text(hjust = 0.5, size = 12),
    legend.position = "bottom"
  )
```



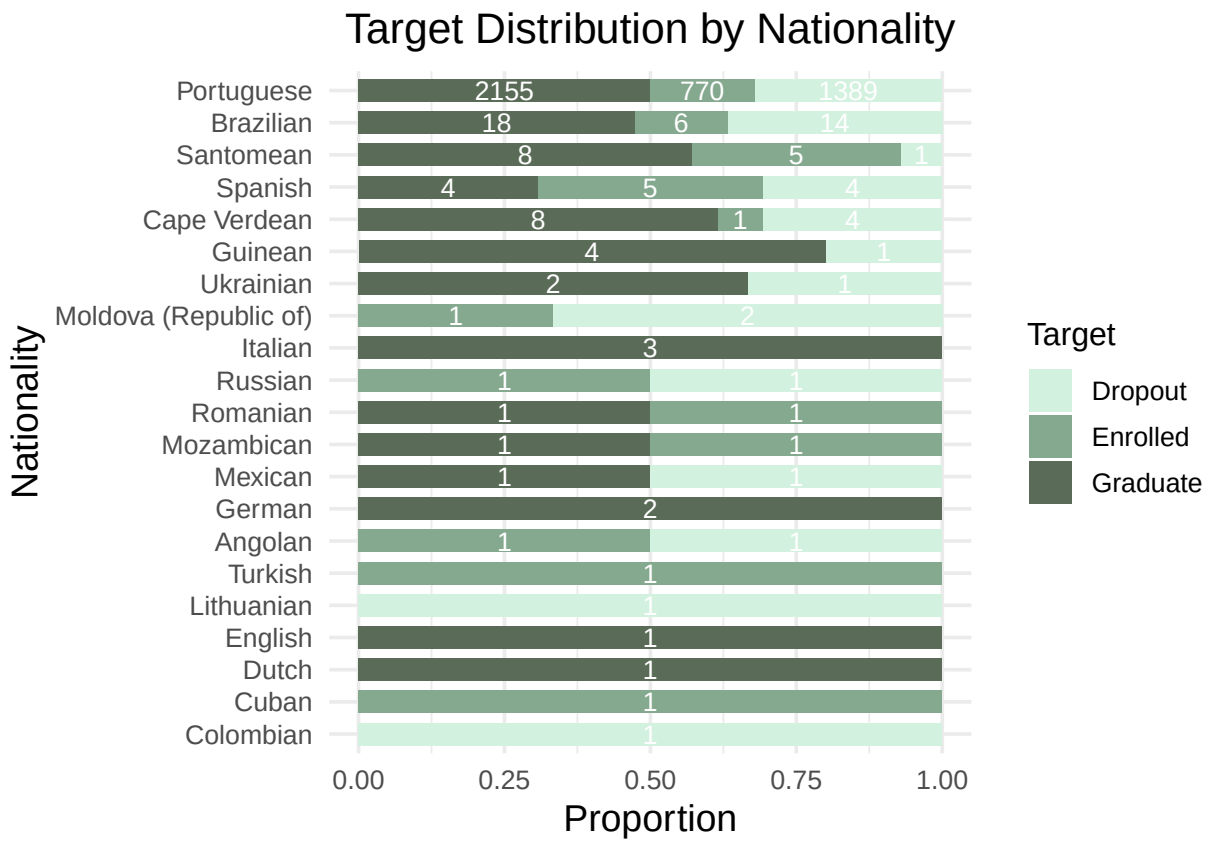

```
library(ggplot2)
library(dplyr)

# Count and extract the top 5 nationalities excluding Portuguese
top_5_nationalities <- data %>%
  count(Nacionality, sort = TRUE) %>%
  filter(Nacionality != "Portuguese") %>%  # Exclude Portuguese
  top_n(5, n)
```

```r
# Plot the top 5 nationalities with counts and limit the y-axis
ggplot(top_5_nationalities, aes(x = reorder(Nacionality, n), y = n, fill = Nacionality)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = n), vjust = -0.3, size = 5) +
  scale_fill_brewer(palette = "Greens") +  # Apply green palette
  ylim(0, max(top_5_nationalities$n) + 50) +  # Set y-axis limit with some padding
  labs(
    title = "Top 5 Nationalities by Number of Students Excluding Portuguese",
    x = "Nationality",
    y = "Number of Students"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.position = "none"
  )
```



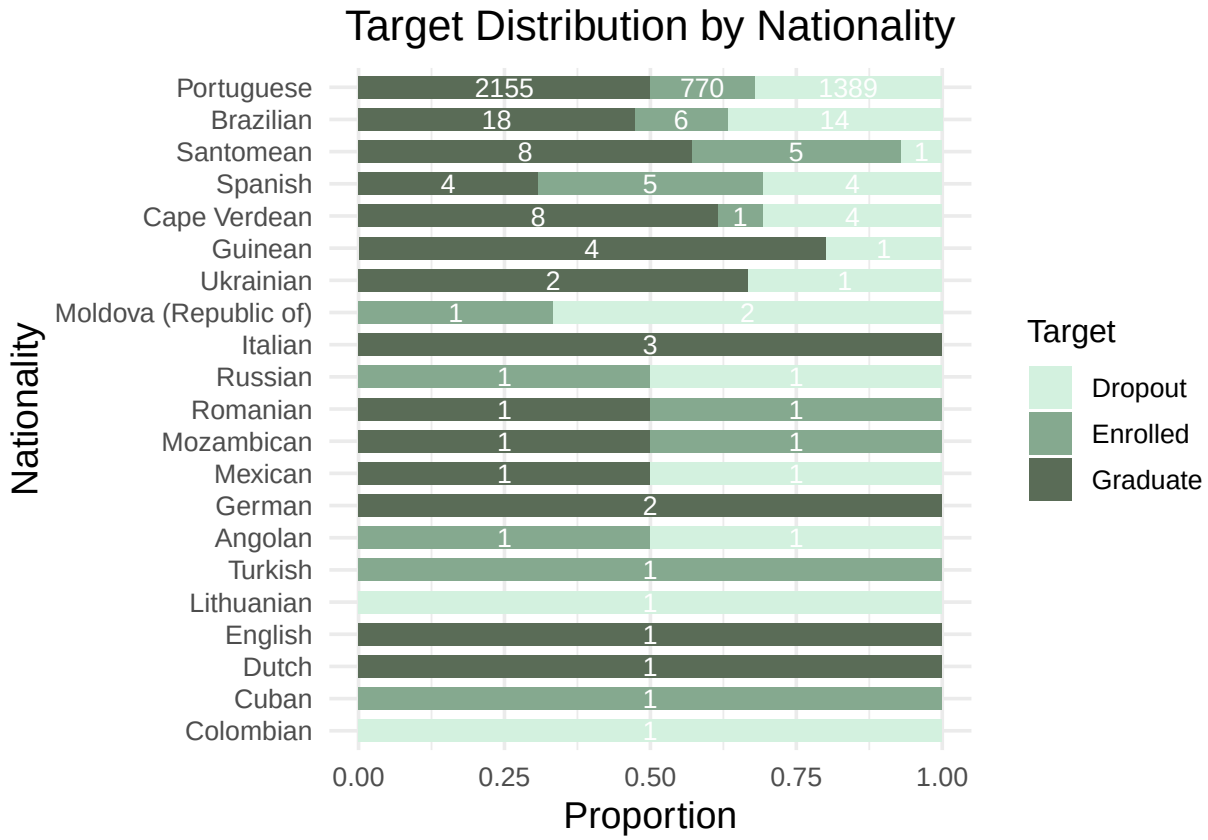

```r
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Aggregate data by Nationality and Target
nationality_target <- data %>%
  group_by(Nacionality, Target) %>%
  summarise(Count = n(), .groups = 'drop')
```

```r
# Reorder nationalities based on total count for better readability
nationality_target <- nationality_target %>%
  group_by(Nacionality) %>%
  mutate(Total = sum(Count)) %>%
  ungroup() %>%
  mutate(Nacionality = reorder(Nacionality, Total))

# Plot for Nationality and Target
ggplot(nationality_target, aes(x = Count, y = Nacionality, fill = Target)) +
  geom_bar(stat = "identity", position = "fill", width = 0.7) +  # Adjust position to "fill" for full-w
  geom_text(
    aes(label = Count),
    position = position_fill(vjust = 0.5),  # Place text in the center of the bars
    color = "white",
    size = 3.5
  ) +
  labs(
    title = "Target Distribution by Nationality",
    x = "Proportion",
    y = "Nationality",
    fill = "Target"
  ) +
  scale_fill_manual(
    values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F")
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12)
  )
```



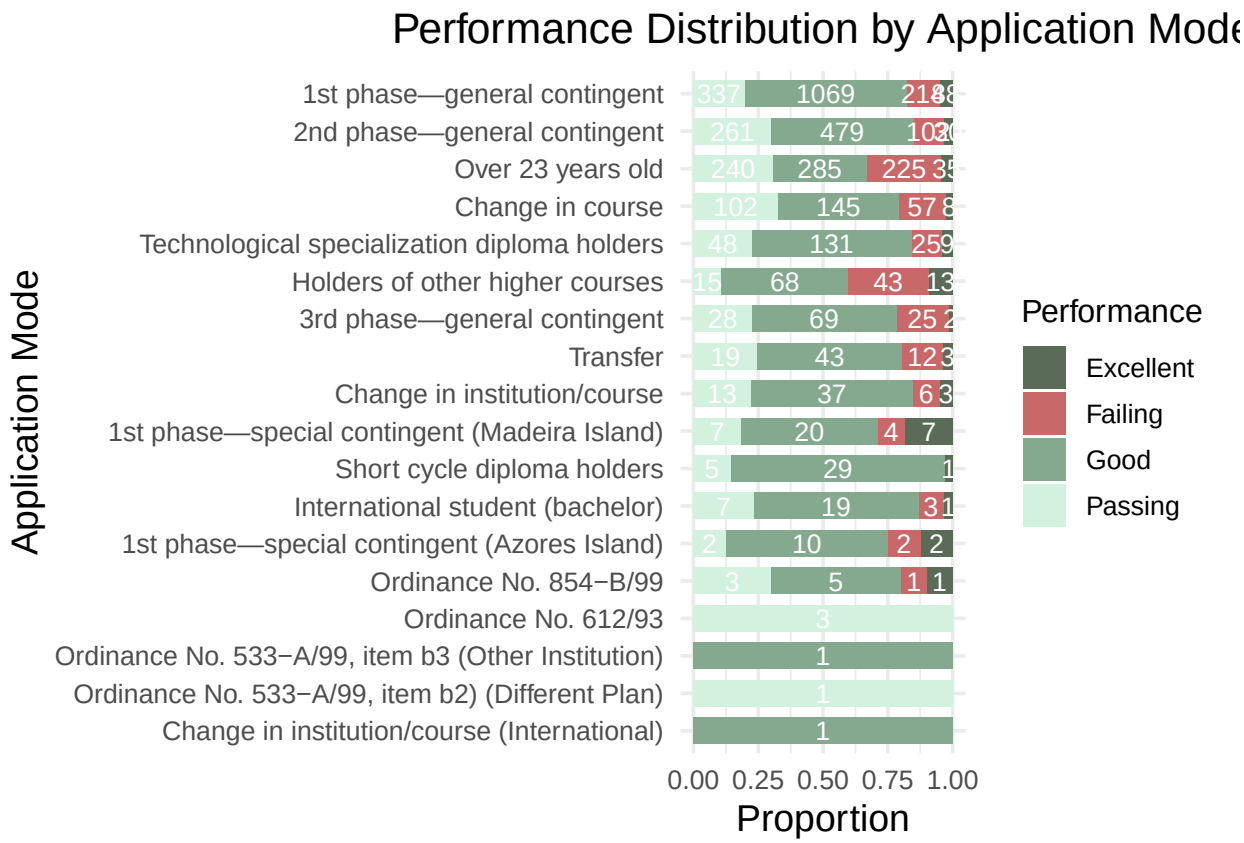

```r
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Aggregate data by Nationality and Target
nationality_target <- data %>%
  group_by(Nacionality, Target) %>%
  summarise(Count = n(), .groups = 'drop')

# Reorder nationalities based on total count for better readability
nationality_target <- nationality_target %>%
  group_by(Nacionality) %>%
  mutate(Total = sum(Count)) %>%
  ungroup() %>%
  mutate(Nacionality = reorder(Nacionality, Total))

# Plot for Nationality and Target
ggplot(nationality_target, aes(x = Count, y = Nacionality, fill = Target)) +
  geom_bar(stat = "identity", position = "fill", width = 0.7) +  # Adjust position to "fill" for full-w
  geom_text(
    aes(label = Count),
    position = position_fill(vjust = 0.5),  # Place text in the center of the bars
    color = "white",
    size = 3.5
  ) +
  labs(
```

```
    title = "Target Distribution by Nationality",
    x = "Proportion",
    y = "Nationality",
    fill = "Target"
  ) +
  scale_fill_manual(
      values = c("Graduate" = "#5A6C57", "Dropout" = "#D3F1DF", "Enrolled" = "#85A98F")
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12)
  )
```




```
# Load necessary libraries
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Aggregate data by Application Mode and Performance
application_performance <- data %>%
  group_by(Application.mode, First.Sem.Performance) %>%
```

```r
  summarise(Count = n(), .groups = 'drop')

# Reorder application modes based on total count for better readability
application_performance <- application_performance %>%
  group_by(Application.mode) %>%
  mutate(Total = sum(Count)) %>%
  ungroup() %>%
  mutate(Application.mode = reorder(Application.mode, Total))

# Plot for Application Mode and Performance
ggplot(application_performance, aes(x = Count, y = Application.mode, fill = First.Sem.Performance)) +
  geom_bar(stat = "identity", position = "fill", width = 0.7) +  # Adjust position to "fill" for full-w
  geom_text(
    aes(label = Count),
    position = position_fill(vjust = 0.5),  # Place text in the center of the bars
    color = "white",
    size = 3.5
  ) +
  labs(
    title = "Performance Distribution by Application Mode",
    x = "Proportion",
    y = "Application Mode",
    fill = "Performance"
  ) +
  scale_fill_manual(
    values = c(
      "Failing" = "#C96868",  # Red
  "Passing" = "#D3F1DF",  # Yellow
  "Good" = "#85A98F",     # Green
  "Excellent" = "#5A6C57"
    )
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(size = 10),
    legend.text = element_text(size = 10),
    legend.title = element_text(size = 12)
  )
```


Performance Distribution by Application Mode
Application Mode
1st phase—general contingent
2nd phase—general contingent
Over 23 years old
Change in course
Technological specialization diploma holders
Holders of other higher courses
3rd phase—general contingent
Transfer
Change in institution/course
1st phase—special contingent (Madeira Island)
Short cycle diploma holders
International student (bachelor)
1st phase—special contingent (Azores Island)
Ordinance No. 854−B/99
Ordinance No. 612/93
Ordinance No. 533−A/99, item b3 (Other Institution)
Ordinance No. 533−A/99, item b2) (Different Plan)
Change in institution/course (International)
337 1069 218
261 479 103
240 285 225 3
102 145 57 8
48 131 25 9
15 68 43 13
28 69 25 2
19 43 12 3
13 37 6 3
7 20 4 7
5 29 1
7 19 3 1
2 10 2 2
3 5 1 1
3
1
1
1
0.00 0.25 0.50 0.75 1.00
Proportion
Performance
Excellent
Failing
Good
Passing