Event Data Analysis

Kunwu Lyu

2024-06-26

Loading Packages

```r
library(dplyr)
library(tidyverse)
library(stringr)
library(lubridate)
library(tidyr)
library(forcats)
library(ggplot2)
library(viridis)
library(viridisLite)
library(purrr)
library(gganimate)
library(ggthemes)
library(leaflet)
library(patchwork)
library(readr)
library(janitor)
library(plotly)
```

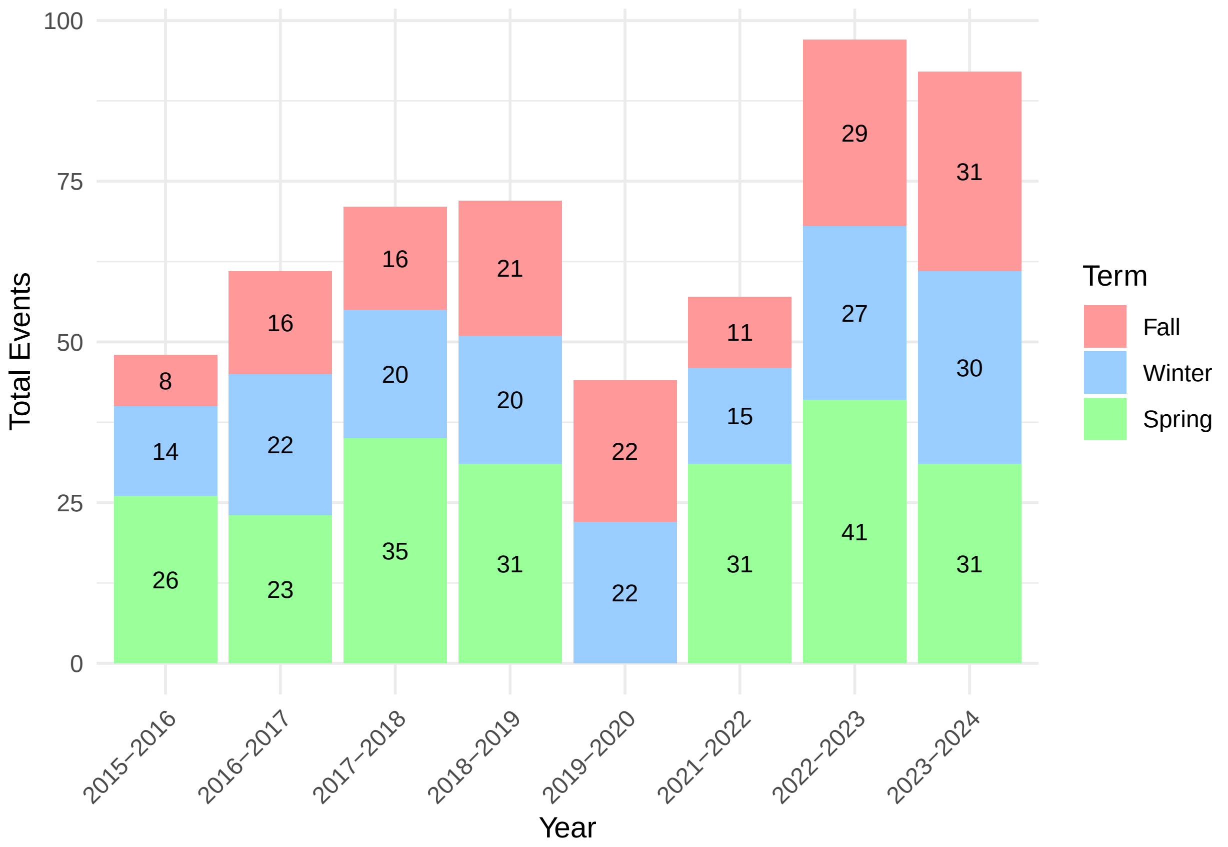
Data Wrangling

```r
# Define the directory containing the files
data_dir <- "data/"

# Define the file codes
file_codes <- c("F15", "W16", "S16", "F16", "W17", "S17", "F17", "W18", "S18", "F18",
                "W19", "S19", "F19", "W20", "S20", "F20", "W21", "S21", "F21", "W22",
                "S22", "F22", "W23", "S23", "F23", "W24", "S24")

# Create a function to read a file given its code
read_and_clean_event_file <- function(code) {
  file_path <- paste0(data_dir, "2015-2024 Events Data - ", code, " - Event Data.csv")
  data <- read_csv(file_path)
  clean_names(data) %>%
    mutate(across(everything(), as.character)) %>%
    mutate(term = code)
}

term_to_year <- function(term) {
  year <- as.numeric(str_sub(term, 2, 3))
  season <- str_sub(term, 1, 1)
  start_year <- if_else(season == "F", 2000 + year, 2000 + year - 1)
  end_year <- start_year + 1
  return(paste0(start_year, "-", end_year))
}

# Use purrr to read all files and store them in a named list
event_data_list <- set_names(map(file_codes, read_and_clean_event_file), file_codes)

combined_data <- reduce(event_data_list, full_join)

combined_data_filtered <- combined_data %>%
  filter(!is.na(what), what != "") %>%
  filter(what != "Choir & Jazz Rehearsal") %>%
  filter(what != "Jazz Rehearsal") %>%
  mutate(date = as.Date(ymd(date))) %>%
  mutate(support_level = if_else(support_level == "N" | support_level == "Y", "L", support_level)) %>%
  mutate(
    department = str_replace_all(department, "WCC", "ODOA"),
    department = str_replace_all(department, "MSUC", "MUSC"),
    department = str_replace_all(department, "French Dept|French", "FREN"),
    department = str_replace_all(department, "English", "ENGL"),
    department = str_replace_all(department, "Pres. Office", "PRES"),
    department = str_replace_all(department, "History", "HIST"),
    department = str_replace_all(department, "THD", "THDA"),
    department = str_replace_all(department, "Inclusion & Equity", "IEC"),
    venue = str_replace_all(venue, "Skinner Chapel", "Chapel"),
    department = str_replace_all(department, "/", " & "),
    department = str_replace_all(department, ",", " &"),
    venue = str_replace_all(venue, ",", " &"),
    department = str_replace_all(department, "\\s+", " ")  # Remove extra spaces
  ) %>%
  select(-wk) %>%
  mutate(department_type = case_when(
      department == "MUSC" ~ "MUSC",
      department == "ODOA" ~ "ODOA",
      department == "CSA" ~ "CSA",
      str_detect(department, "&") ~ "Collab",
      TRUE ~ "Others"
    )) %>%
  mutate(what = str_replace_all(what, "Jazz Ensemble Concert|Jazz Area Concert", "Jazz Concert"),
         what = str_replace_all(what, "Symphony Band Concert", "Symphony Concert"),
         what = str_replace_all(what, "Composition Recital", "Composition Showcase Recital"),
         what = str_replace_all(what, "Harpichord", "Harpsichord"),
         what = str_replace_all(what, "Emsemble", "Ensemble"),
         what = str_replace_all(what, "Juest Cellin'", "Just Cellin'"),
         what = str_replace_all(what, "Facutly|FACULTY|Mazariello", "Faculty")) %>%
  mutate(event_type = case_when(
    str_detect(what, "GUEST|ODOA|Concert Series|SPCO") ~ "Guest",
    str_detect(what, "Faculty") ~ "Faculty Recital",
    str_detect(what, "Student|Senior|Junior|Piano Recital: |Johnson|Verma Jameson") ~ "Student Recital"
    str_detect(what, "Studio Recital|Organ & Harpsichord|Composition Showcase Recital|Chamber Recital|C
    str_detect(what, "Orchestra Concert|Jazz Concert|Symphony Concert|Symphony Band|Choir Concert|Orche
    str_detect(what, "CSA|Just Cellin|Lunar New Year|ACA|A Cappella|Accidentals|Exit 69|Date Knight|Kni
    str_detect(what, "Masterclass|Lecture|Symposium") ~ "Masterclass",
    str_detect(what, "Trustees|Trustee's|Presidents|Conference|President's|Presentation") ~ "Presentatio
    str_detect(what, "Clinic|Music Fest|Music Department Showcase|Melinda Russell|Launch|Event|Opening"
      TRUE ~ "Guest"
  )) %>%
  mutate(year = term_to_year(term)) %>%
  mutate(term = factor(term, levels = c("F15", "W16", "S16", "F16", "W17", "S17",
                                        "F17", "W18", "S18", "F18", "W19", "S19",
                                        "F19", "W20", "S20", "F20", "W21", "S21",
                                        "F21", "W22", "S22", "F22", "W23", "S23",
                                        "F23", "W24", "S24"), ordered = TRUE)) %>%
  arrange(year, term) %>%
  mutate(
    term_category = case_when(
      str_detect(term, "^F") ~ "Fall",
      str_detect(term, "^W") ~ "Winter",
      str_detect(term, "^S") ~ "Spring"
    )
  ) %>%
  mutate(term_category = factor(term_category, levels = c("Spring", "Winter", "Fall"), ordered = TRUE))

glimpse(combined_data_filtered)
Rows: 542
Columns: 24
$ venue           <chr> "Concert Hall", "Concert Hall", "Concert Hall", "Conce~
$ date            <date> 2015-10-25, 2015-10-30, 2015-11-07, 2015-11-08, 2015--
$ what            <chr> "Jazz Concert", "Symphony Concert", "Choir Concert", "~
$ department      <chr> "MUSC", "MUSC", "MUSC", "MUSC", "MUSC", "CSA", "MUSC",~
$ days_committed  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ av_staff        <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ pac_staff       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ support_level   <chr> "H", "M", "M", "H", "H", "L", "M", "L", "M", "L", "L",~
$ audio_needs     <chr> "H", "L", "L", "M", "L", "L", "L", "L", "M", "L", "L",~
$ stage_needs     <chr> "H", "H", "H", "H", "H", "M", "M", NA, "M", "L", "L", ~
$ lighting_needs  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ projection      <chr> "N", "Y", "N", "Y", "N", "N", "N", NA, "N", "N", "N", ~
$ video_recording <chr> "N", "N", "Y", "Y", "Y", "Y", "N", NA, "Y", "N", "Y", ~
$ live_stream     <chr> "N", "N", "N", "N", "N", "N", "N", "N", NA, NA, NA, NA~
$ audience_count  <chr> "64", "122", "86", "77", "220", "192", "44", NA, "137"~
$ poster          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ program         <chr> "Y", "Y", "Y", "Y", "Y", "N", "Y", "N", "Y", "Y", "Y",~
$ reception       <chr> "Y", "Y", "Y", "Y", "Y", "N", "N", "N", "Y", "Y", "Y",~
$ term            <ord> F15, F15, F15, F15, F15, F15, F15, F15, W16, W16, W16,~
$ livestream      <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ department_type <chr> "MUSC", "MUSC", "MUSC", "MUSC", "MUSC", "CSA", "MUSC",~
$ event_type      <chr> "Ensemble Concert", "Ensemble Concert", "Ensemble Conc~
$ year            <chr> "2015-2016", "2015-2016", "2015-2016", "2015-2016", "2~
$ term_category   <ord> Fall, Fall, Fall, Fall, Fall, Fall, Fall, Fall, Winter~


event_summary <- combined_data_filtered %>%
  group_by(year, term) %>%
  summarize(term_total = n(), .groups = 'drop') %>%
  group_by(year) %>%
  mutate(year_total = sum(term_total)) %>%
  ungroup()
```

EDA Plots

Overall Event Summary

```
# Sort the data frame by year and term in the order of F, W, S
event_summary <- event_summary %>%
  mutate(term = factor(term, levels = c("F15", "W16", "S16", "F16", "W17", "S17",
                                        "F17", "W18", "S18", "F18", "W19", "S19",
                                        "F19", "W20", "S20", "F20", "W21", "S21",
                                        "F21", "W22", "S22", "F22", "W23", "S23",
                                        "F23", "W24", "S24"), ordered = TRUE)) %>%
  arrange(year, term) %>%
  mutate(
    term_category = case_when(
      str_detect(term, "^F") ~ "Fall",
      str_detect(term, "^W") ~ "Winter",
      str_detect(term, "^S") ~ "Spring"
    )
  ) %>%
  mutate(term_category = factor(term_category, levels = c("Fall", "Winter", "Spring"), ordered = TRUE))


# Create the stacked bar chart
ggplot(event_summary, aes(x = year, y = term_total, fill = term_category)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = term_total),
            position = position_stack(vjust = 0.5),
            size = 3,
            color = "black") +
  scale_fill_manual(values = c("Fall" = "#FF9999", "Winter" = "#99CCFF", "Spring" = "#99FF99")) +
  labs(x = "Year", y = "Total Events", fill = "Term") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

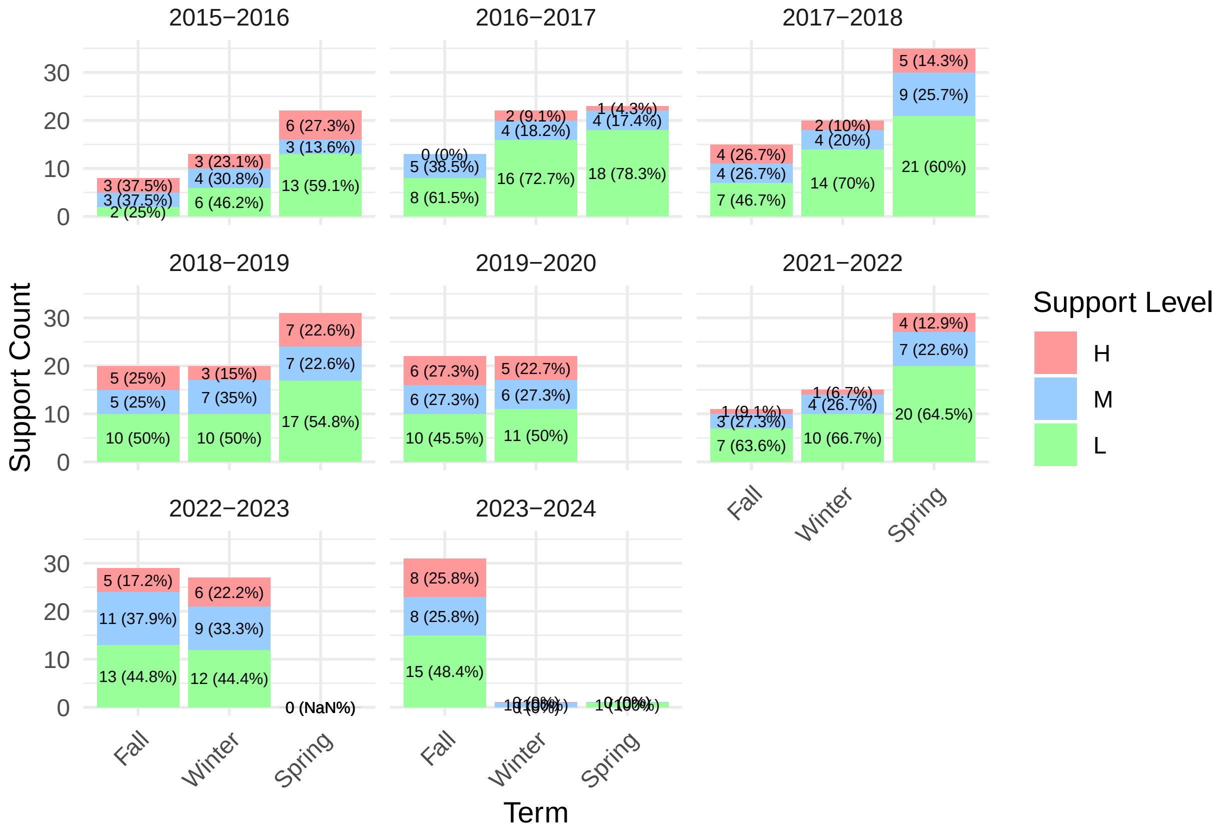


```
event_summary
# A tibble: 23 x 5
   year      term  term_total year_total term_category
   <chr>     <ord>      <int>      <int> <ord>
 1 2015-2016 F15            8         48 Fall
 2 2015-2016 W16           14         48 Winter
 3 2015-2016 S16           26         48 Spring
 4 2016-2017 F16           16         61 Fall
 5 2016-2017 W17           22         61 Winter
 6 2016-2017 S17           23         61 Spring
 7 2017-2018 F17           16         71 Fall
 8 2017-2018 W18           20         71 Winter
 9 2017-2018 S18           35         71 Spring
10 2018-2019 F18           21         72 Fall
# i 13 more rows
```

Breakdown of Events by Support Level

```
library(dplyr)
library(tidyr)
library(purrr)
library(ggplot2)

# Function to summarize and pivot data for each year
```

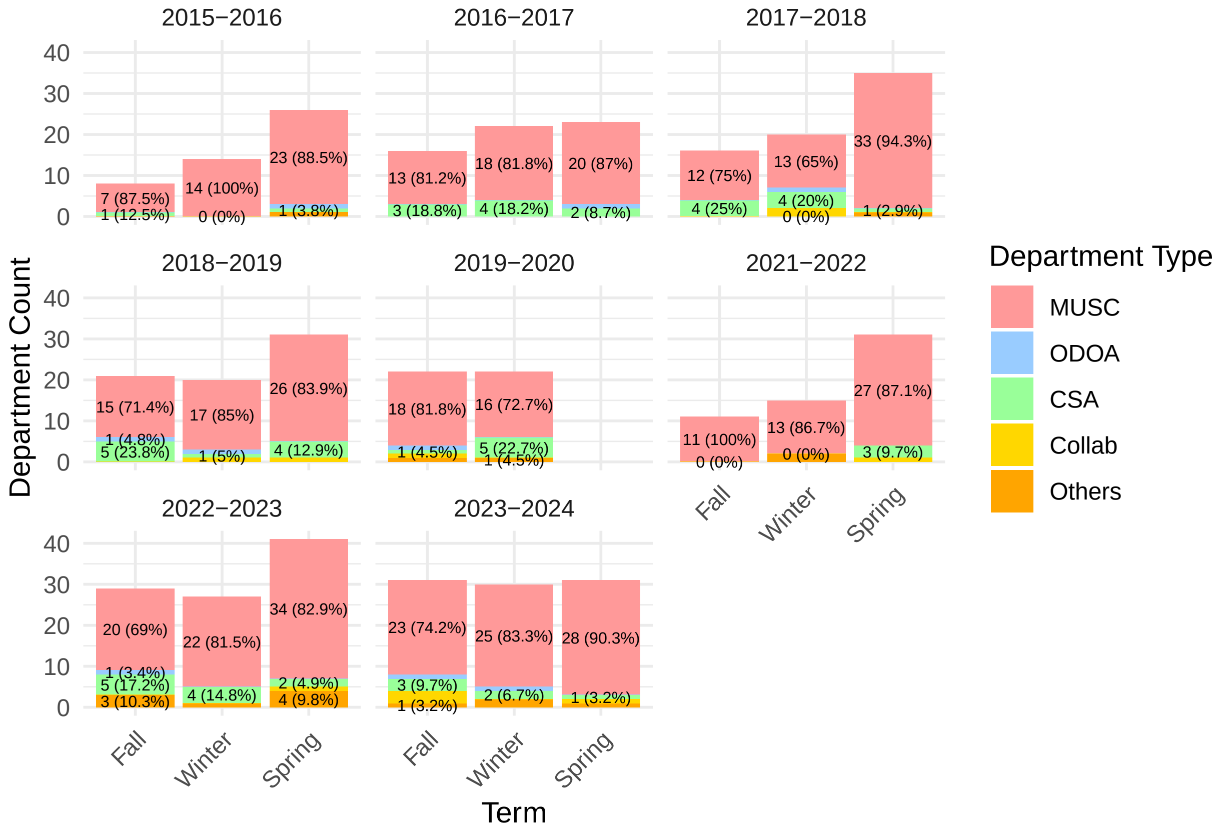
```r
summarize_and_pivot <- function(current_year) {
  combined_data_filtered %>%
    filter(year == current_year) %>%
    group_by(term_category, support_level) %>%
    summarize(Support = n(), .groups = 'drop') %>%
    pivot_wider(names_from = support_level, values_from = Support, values_fill = list(Support = 0)) %>%
    mutate(year = current_year)  # Add a year column to identify the table
}

# List of years to iterate over
years <- unique(combined_data_filtered$year)

# Use purrr::map to apply summarize_and_pivot function for each year
support_tables <- map_dfr(years, summarize_and_pivot)  # Combine into a single data frame

# Melt the data for ggplot
support_tables_melted <- support_tables %>%
  pivot_longer(cols = -c(term_category, year), names_to = "support_level", values_to = "Support")

# Ensure the term_category and support_level are factors with the correct order
support_tables_melted <- support_tables_melted %>%
  filter(support_level != "NA") %>%
  mutate(term_category = factor(term_category, levels = c("Fall", "Winter", "Spring"), ordered = TRUE),
         support_level = factor(support_level, levels = c("NA", "H", "M", "L"), ordered = TRUE))

# Calculate percentages for each segment
support_tables_melted <- support_tables_melted %>%
  group_by(year, term_category) %>%
  mutate(total_support = sum(Support),
         percentage = (Support / total_support) * 100)

# Create the facet plot
ggplot(support_tables_melted, aes(x = term_category, y = Support, fill = support_level)) +
  geom_bar(stat = "identity", position = "stack") +
  geom_text(aes(label = paste0(Support, " (", round(percentage, 1), "%)")),
            position = position_stack(vjust = 0.5), size = 2, color = "black") +
  facet_wrap(~ year) +
  labs(x = "Term", y = "Support Count", fill = "Support Level") +
  scale_fill_manual(values = c("H" = "#FF9999", "M" = "#99CCFF", "L" = "#99FF99", "NA" = "black")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
support_tables_melted
# A tibble: 69 x 6
# Groups:   year, term_category [23]
   term_category year      support_level Support total_support percentage
   <ord>         <chr>     <ord>           <int>         <int>      <dbl>
 1 Spring        2015-2016 H                   6            22      27.3
 2 Spring        2015-2016 L                  13            22      59.1
 3 Spring        2015-2016 M                   3            22      13.6
 4 Winter        2015-2016 H                   3            13      23.1
 5 Winter        2015-2016 L                   6            13      46.2
 6 Winter        2015-2016 M                   4            13      30.8
 7 Fall          2015-2016 H                   3             8      37.5
 8 Fall          2015-2016 L                   2             8      25
 9 Fall          2015-2016 M                   3             8      37.5
10 Spring        2016-2017 H                   1            23       4.35
# i 59 more rows
```

Breakdown of Events by Department/Source

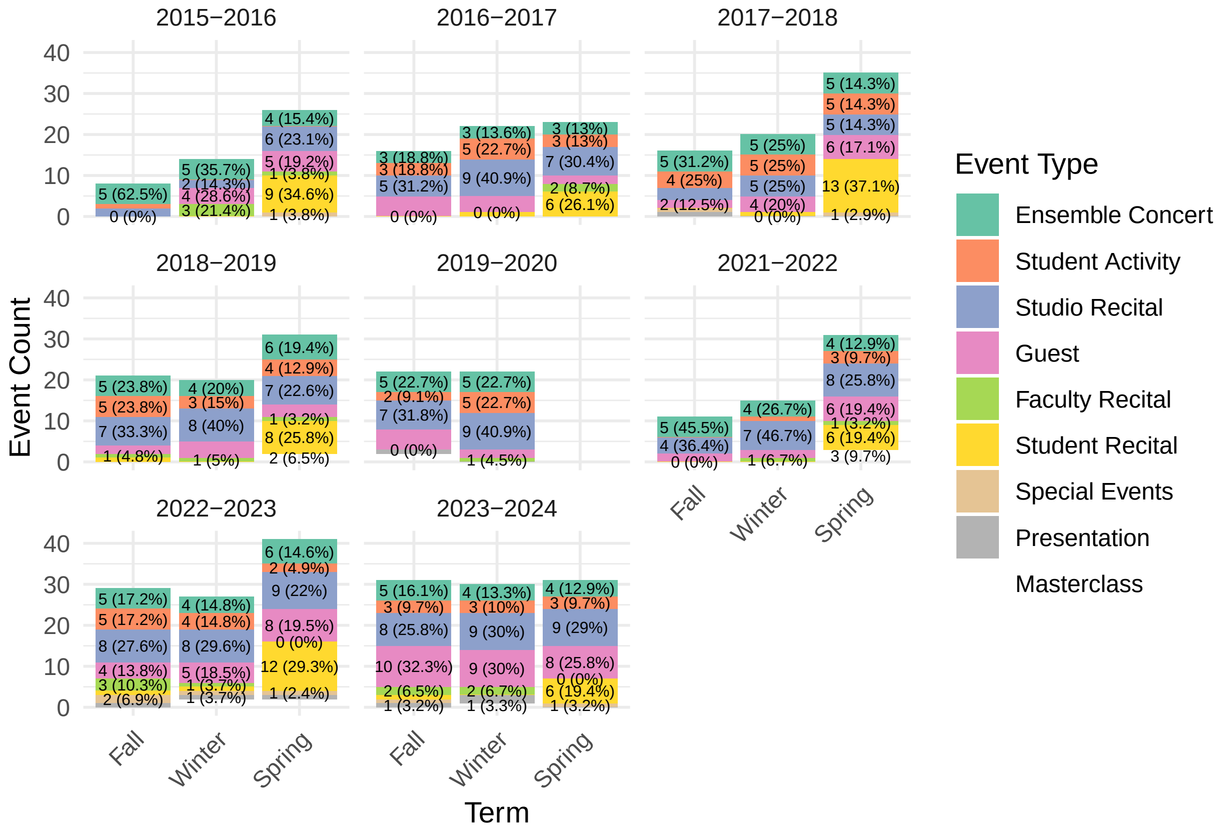
```
# Function to summarize and pivot data for each year
summarize_and_pivot_department <- function(current_year) {
  combined_data_filtered %>%
    filter(year == current_year) %>%
    group_by(term_category, department_type) %>%
    summarize(DepartmentCount = n(), .groups = 'drop') %>%
    pivot_wider(names_from = department_type, values_from = DepartmentCount, values_fill = list(Departm
    mutate(year = current_year)  # Add a year column to identify the table
}

# List of years to iterate over
years <- unique(combined_data_filtered$year)

# Use purrr::map to apply summarize_and_pivot function for each year
department_tables <- map_dfr(years, summarize_and_pivot_department)  # Combine into a single data frame

# Melt the data for ggplot
department_tables_melted <- department_tables %>%
  pivot_longer(cols = -c(term_category, year), names_to = "department_type", values_to = "DepartmentCou

# Ensure the term_category and department_type are factors with the correct order
department_tables_melted <- department_tables_melted %>%
  mutate(term_category = factor(term_category, levels = c("Fall", "Winter", "Spring"), ordered = TRUE),
         department_type = factor(department_type, levels = c("MUSC", "ODOA", "CSA", "Collab", "Others"

# Filter out NA department types (if any)
department_tables_melted <- department_tables_melted %>%
  filter(department_type != "NA")

# Calculate percentages for each segment
department_tables_melted <- department_tables_melted %>%
  group_by(year, term_category) %>%
  mutate(total_count = sum(DepartmentCount, na.rm = T),
         percentage = (DepartmentCount / total_count) * 100)

# Create the facet plot
ggplot(department_tables_melted, aes(x = term_category, y = DepartmentCount, fill = department_type)) +
  geom_bar(stat = "identity", position = "stack") +
  geom_text(aes(label = paste0(DepartmentCount, " (", round(percentage, 1), "%)")),
            position = position_stack(vjust = 0.5), size = 2, color = "black",
            check_overlap = TRUE) +
  facet_wrap(~ year) +
  labs(x = "Term", y = "Department Count", fill = "Department Type") +
  scale_fill_manual(values = c("MUSC" = "#FF9999", "ODOA" = "#99CCFF", "CSA" = "#99FF99", "Collab" = "#
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
department_tables_melted
# A tibble: 115 x 6
# Groups:   year, term_category [23]
   term_category year     department_type DepartmentCount total_count percentage
   <ord>         <chr>    <ord>                     <int>       <int>      <dbl>
 1 Spring        2015-20~ CSA                           1          26       3.85
 2 Spring        2015-20~ MUSC                         23          26      88.5
 3 Spring        2015-20~ ODOA                          1          26       3.85
 4 Spring        2015-20~ Others                        1          26       3.85
 5 Spring        2015-20~ Collab                       NA          26      NA
 6 Winter        2015-20~ CSA                           0          14       0
 7 Winter        2015-20~ MUSC                         14          14     100
 8 Winter        2015-20~ ODOA                          0          14       0
 9 Winter        2015-20~ Others                        0          14       0
10 Winter        2015-20~ Collab                       NA          14      NA
# i 105 more rows
```

Breakdown of Music, Collab & ODOA Events by Type

```
# Function to summarize and pivot data for each year
summarize_and_pivot_event <- function(current_year) {
  combined_data_filtered %>%
    filter(year == current_year) %>%
    group_by(term_category, event_type) %>%
    summarize(EventCount = n(), .groups = 'drop') %>%
    pivot_wider(names_from = event_type, values_from = EventCount, values_fill = list(EventCount = 0)) %
    mutate(year = current_year)  # Add a year column to identify the table
}

# List of years to iterate over
years <- unique(combined_data_filtered$year)

# Use purrr::map to apply summarize_and_pivot function for each year
event_tables <- map_dfr(years, summarize_and_pivot_event)  # Combine into a single data frame

# Melt the data for ggplot
event_tables_melted <- event_tables %>%
  pivot_longer(cols = -c(term_category, year), names_to = "event_type", values_to = "EventCount")

# Ensure the term_category and event_type are factors with the correct order
event_tables_melted <- event_tables_melted %>%
  mutate(term_category = factor(term_category, levels = c("Fall", "Winter", "Spring"), ordered = TRUE),
         event_type = factor(event_type, levels = c("Ensemble Concert", "Student Activity", "Studio Reci
                                                    "Guest", "Faculty Recital", "Student Recital",
                                                    "Special Events", "Presentation", "Masterclass"),
                            ordered = TRUE))

# Calculate percentages for each segment
event_tables_melted <- event_tables_melted %>%
  group_by(year, term_category) %>%
  mutate(total_count = sum(EventCount, na.rm = T),
         percentage = (EventCount / total_count) * 100)

# Create the facet plot
ggplot(event_tables_melted, aes(x = term_category, y = EventCount, fill = event_type)) +
  geom_bar(stat = "identity", position = "stack") +
  geom_text(aes(label = paste0(EventCount, " (", round(percentage, 1), "%)")),
            position = position_stack(vjust = 0.5), size = 2, color = "black",
            check_overlap = TRUE) +
  facet_wrap(~ year) +
  labs(x = "Term", y = "Event Count", fill = "Event Type") +
  scale_fill_brewer(palette = "Set2") +  # Using a Brewer palette for default colors
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
event_tables_melted
# A tibble: 207 x 6
# Groups:   year, term_category [23]
   term_category year      event_type       EventCount total_count percentage
   <ord>         <chr>     <ord>                 <int>       <int>      <dbl>
 1 Spring        2015-2016 Ensemble Concert          4          26      15.4
 2 Spring        2015-2016 Faculty Recital           1          26       3.85
 3 Spring        2015-2016 Guest                     5          26      19.2
 4 Spring        2015-2016 Special Events            1          26       3.85
 5 Spring        2015-2016 Student Recital           9          26      34.6
 6 Spring        2015-2016 Studio Recital            6          26      23.1
 7 Spring        2015-2016 Student Activity          0          26       0
 8 Spring        2015-2016 Presentation             NA          26      NA
 9 Spring        2015-2016 Masterclass              NA          26      NA
10 Winter        2015-2016 Ensemble Concert          5          14      35.7
# i 197 more rows
```