datathon_analysis

Kaiwen Ren

2024-03-23

```
data <- read.csv("/Users/kelsey/Downloads/cleaned_data_latest.csv")
```

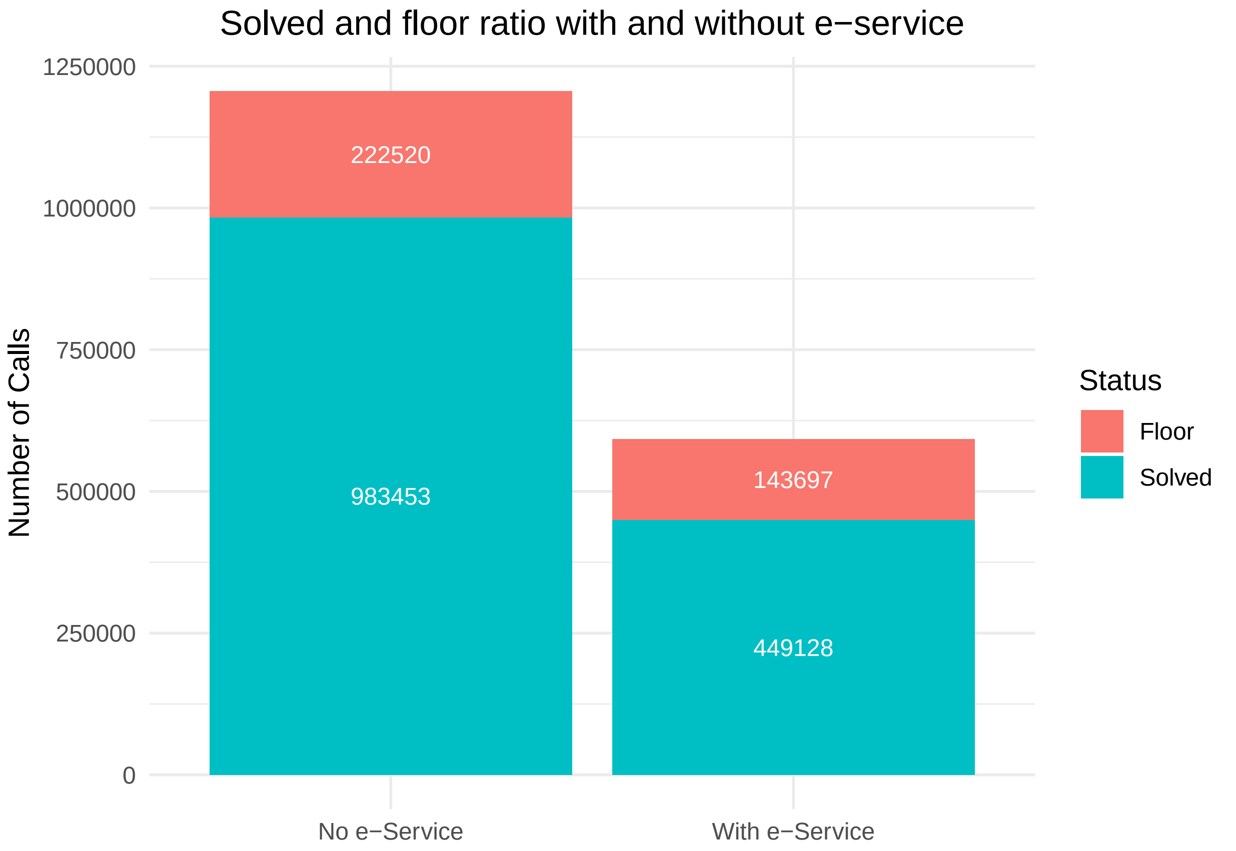
No e-service VS e-service resolved percentage

```
result <- data %>%
  mutate(combination = interaction(resolved, eservice_ind_13_march)) %>%
  count(combination) %>%
  spread(key = combination, value = n, fill = 0) %>%
  mutate(
    percent_resolved_eservice = `1.1` / (`1.1` + `0.1`) * 100,
    percent_resolved_no_eservice = `1.0` / (`1.0` + `0.0`) * 100,
    percent_floor_eservice = `0.1` / (`1.1` + `0.1`) * 100,
    percent_floor_no_eservice = `0.0` / (`1.0` + `0.0`) * 100
  )

print(result)
```

```
##      0.0    1.0    0.1    1.1 percent_resolved_eservice
## 1 222520 983453 143697 449128                  75.76064
##   percent_resolved_no_eservice percent_floor_eservice percent_floor_no_eservice
## 1                     81.54851               24.23936                  18.45149
```

Solved and floor ration with and without e-service

```
# Create a data frame with your data
data <- data.frame(
  e_service = rep(c("With e-Service", "No e-Service"), each = 2),
  status = rep(c("Floor", "Solved"), times = 2),
  calls = c(143697, 449128, 222520, 983453)  # Reordered to match the new status order
)

# Arrange the data to make 'Floor' come on top of 'Solved' in the plot
data$status <- factor(data$status, levels = c("Floor", "Solved"))

# Plot
ggplot(data, aes(x = e_service, y = calls, fill = status)) +
  geom_col(position = "stack") +
  geom_text(aes(label = calls), position = position_stack(vjust = 0.5), color = "white", size = 3) +
  labs(
    x = NULL,
    y = "Number of Calls",
    fill = "Status",
    title = "Solved and floor ratio with and without e-service",
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

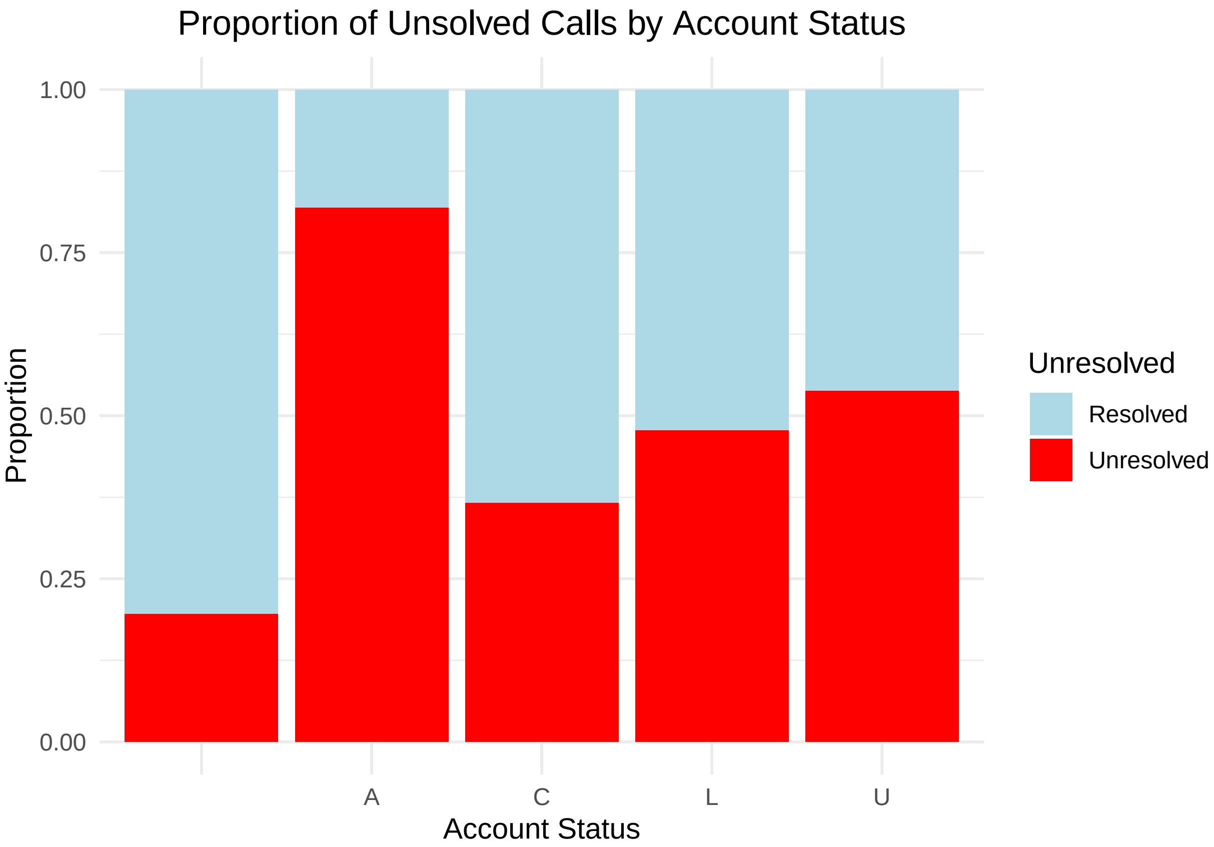


Separate the mos

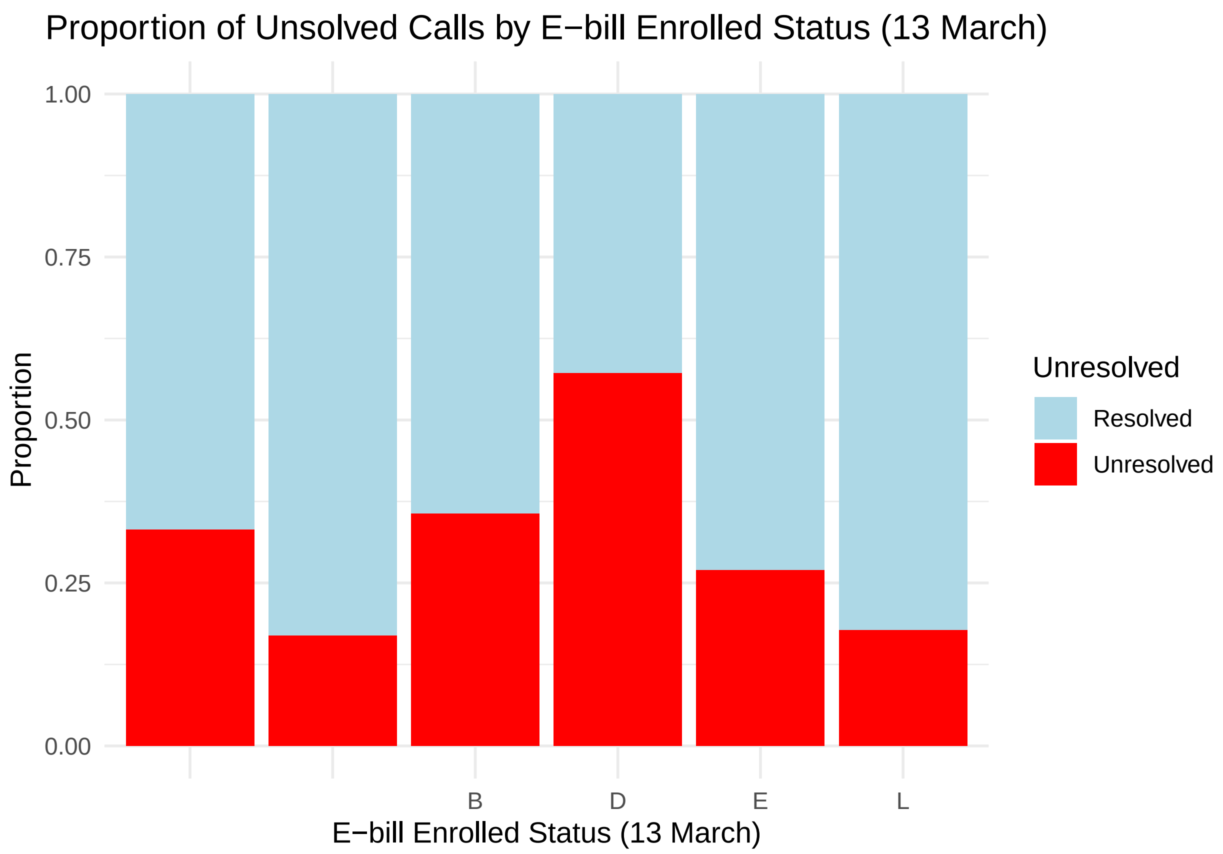
```
converted_data <- read.csv("/Users/kelsey/Downloads/cleaned_data_latest.csv")
```

Proportion of Unsolved Calls by Account Status

```
converted_data$Unresolved <- ifelse(converted_data$resolved == 0, "Unresolved", "Resolved")

# Plot for Account Status
converted_data$account_status_13_march <- factor(converted_data$account_status_13_march)
ggplot(converted_data, aes(x = account_status_13_march, fill = Unresolved)) +
  geom_bar(position = "fill") +
  labs(x = "Account Status", y = "Proportion") +
  theme_minimal() +
  scale_fill_manual(values = c("Resolved" = "lightblue", "Unresolved" = "red")) +
  ggtitle("Proportion of Unsolved Calls by Account Status") +
  theme(plot.title = element_text(hjust = 0.5))
```



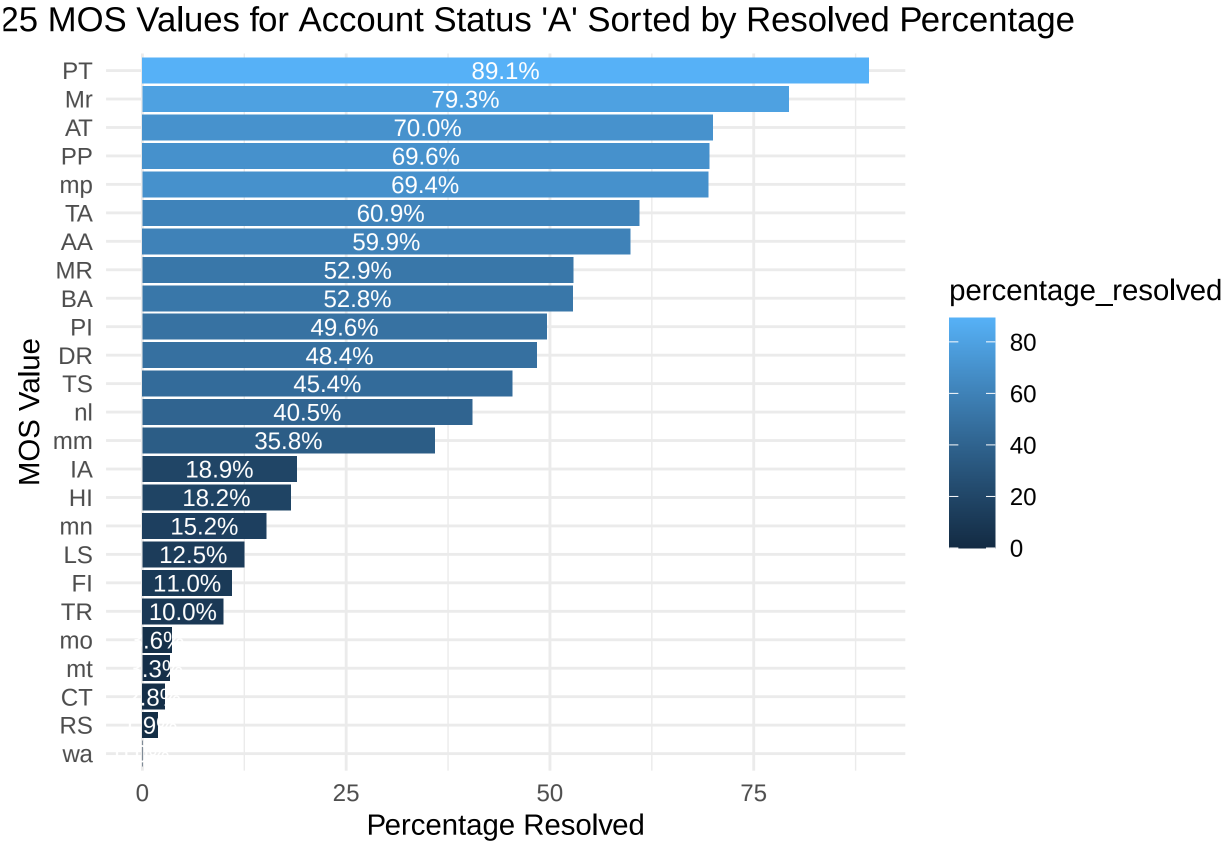
```r
# Plot for E-bill Enrolled Status as of 13 March
ggplot(converted_data, aes(x = ebill_enrolled_status_13_march, fill = Unresolved)) +
  geom_bar(position = "fill") +
  labs(x = "E-bill Enrolled Status (13 March)", y = "Proportion") +
  theme_minimal() +
  scale_fill_manual(values = c("Resolved" = "lightblue", "Unresolved" = "red")) +
  ggtitle("Proportion of Unsolved Calls by E-bill Enrolled Status (13 March)") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
separated_data <- converted_data %>%
  separate_rows(mos, sep = " ")

percentage_resolved <- separated_data %>%
  filter(account_status_13_march %in% c('A')) %>%
  group_by(mos) %>%
  summarise(
    total_count = n(),  # Total number of cases for this card_activation_status
    solved_count = sum(resolved == 1, na.rm = TRUE),  # Number of solved cases
    percentage_resolved = (solved_count / total_count) * 100  # Calculate the percentage
  ) %>%
  ungroup()  # Remove the grouping

# View the result
print(percentage_resolved)
```

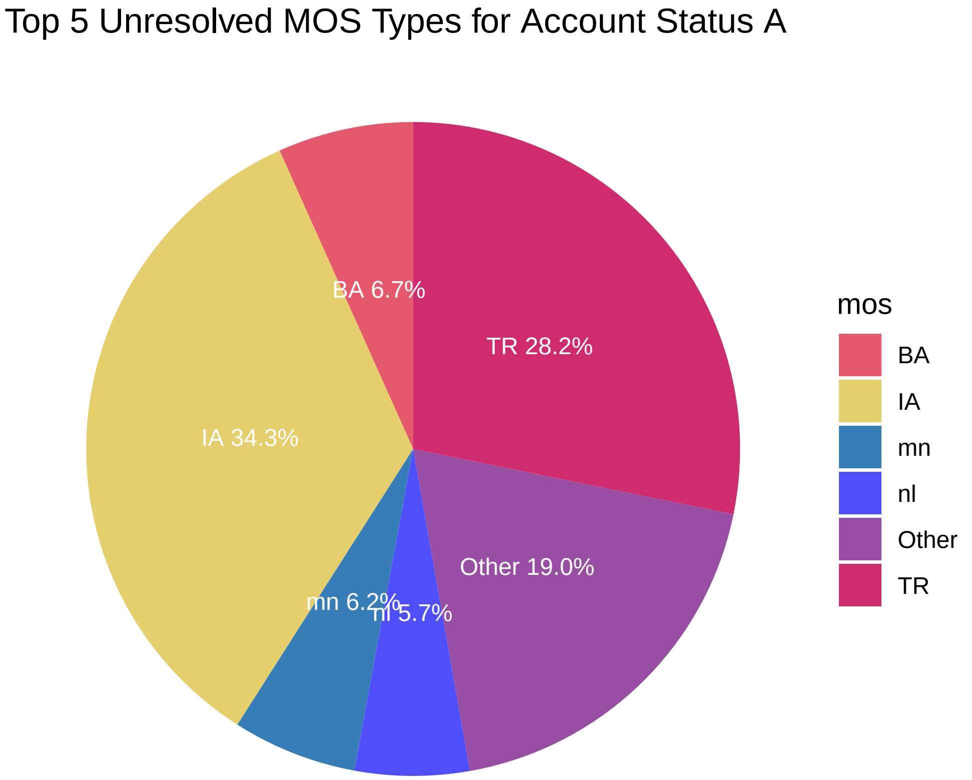
```
## # A tibble: 63 x 4
##    mos   total_count solved_count percentage_resolved
##    <chr>       <int>        <int>               <dbl>
##  1 AA            162           97                59.9
##  2 AP             19            4                21.1
##  3 AT             60           42                70
##  4 AU              1            1               100
##  5 BA           1337          706                52.8
##  6 BL              1            0                 0
##  7 Ba              6            1                16.7
##  8 CA              5            4                80
##  9 CB                14              9                64.3
## 10 CT                72              2                 2.78
## # i 53 more rows
```

```
top_resolved <- percentage_resolved %>%
  slice_max(order_by = total_count, n = 25) %>%
  arrange(percentage_resolved)

ggplot(top_resolved, aes(x = reorder(mos, percentage_resolved), y = percentage_resolved, fill = percenta
  geom_col() +
  geom_text(aes(label = sprintf("%.1f%%", percentage_resolved)), position = position_stack(vjust = 0.5)
  labs(
    x = "MOS Value",
    y = "Percentage Resolved",
    title = "Top 25 MOS Values for Account Status 'A' Sorted by Resolved Percentage"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  coord_flip()
```



```
# Step 1: Filter the data for account_status_13_march 'A' and calculate counts
filtered_data <- separated_data %>%
  filter(account_status_13_march %in% c('A')) %>%
  count(mos)

# Step 2: Identify the top 5 mos types based on count
top_5_mos <- filtered_data %>%
  top_n(5, wt = n)
```

```r
# Step 3: Create a dataset with an "Other" category for MOS types outside the top 5
pie_data <- filtered_data %>%
  mutate(mos = if_else(mos %in% top_5_mos$mos, as.character(mos), "Other")) %>%
  group_by(mos) %>%
  summarise(count = sum(n)) %>%
  mutate(percentage = count / sum(count) * 100) %>%
  ungroup()  # Ensure that the data is no longer grouped for plotting

# Define colors for the top 5 MOS types plus "Other", adjust the number of colors accordingly
pie_chart_colors <- c("#E45A6C", "#E4cf6c", "#377EB8", "#4f4FfA", "#984EA3", "#cF2c6f")

# Step 4: Create the pie chart
ggplot(pie_data, aes(x = "", y = percentage, fill = mos)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  scale_fill_manual(values = pie_chart_colors) +
  theme_void() +
  geom_text(aes(label = paste(mos, sprintf("%.1f%%", percentage))),
            position = position_stack(vjust = 0.5),
            color = "white", size = 3) +
  labs(title = "Top 5 Unresolved MOS Types for Account Status A") +
  theme(legend.position = "right")
```



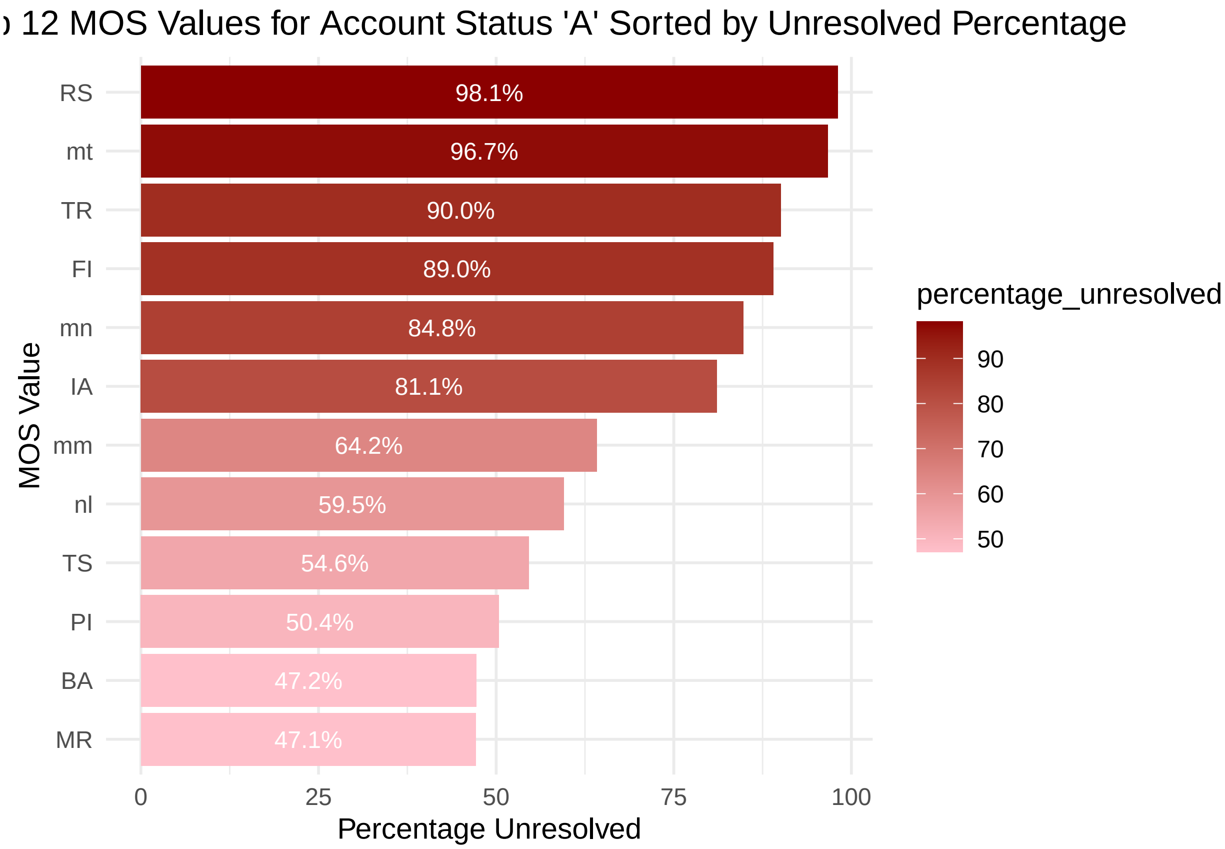
```r
percentage_resolved <- separated_data %>%
  filter(account_status_13_march %in% c('A')) %>%
  group_by(mos) %>%
  summarise(
    total_count = n(),  # Total number of cases for this card_activation_status
    solved_count = sum(resolved == 1, na.rm = TRUE),  # Number of resolved cases
    unsolved_count = total_count - solved_count,  # Number of unresolved cases
    percentage_resolved = (solved_count / total_count) * 100,  # Calculate the percentage resolved
    percentage_unresolved = (unsolved_count / total_count) * 100  # Calculate the percentage unresolved
  ) %>%
  ungroup()  # Remove the grouping

# Debug: Check the intermediate values for a specific 'mos' type
cat("Debug Info for a specific MOS type:\n")
```

```
## Debug Info for a specific MOS type:
```

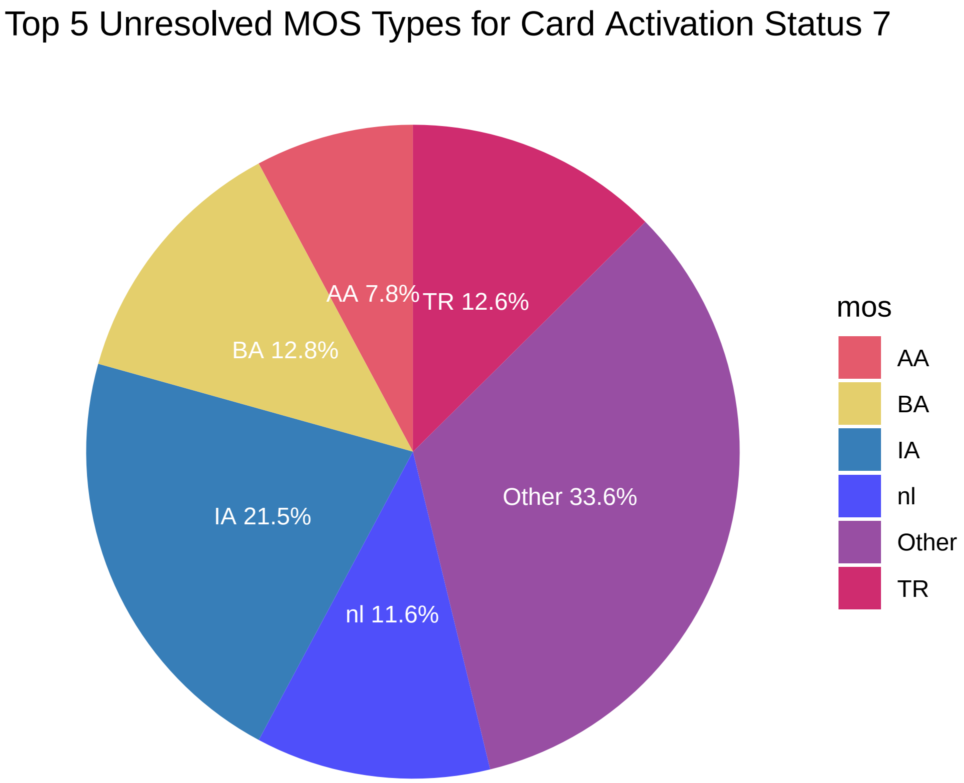
```r
print(percentage_resolved)
```

```
## # A tibble: 63 x 6
##    mos   total_count solved_count unsolved_count percentage_resolved
##    <chr>       <int>        <int>          <int>               <dbl>
##  1 AA            162           97             65               59.9
##  2 AP             19            4             15               21.1
##  3 AT             60           42             18               70
##  4 AU              1            1              0              100
##  5 BA           1337          706            631               52.8
##  6 BL              1            0              1                0
##  7 Ba              6            1              5               16.7
##  8 CA              5            4              1               80
##  9 CB             14            9              5               64.3
## 10 CT             72            2             70                2.78
## # i 53 more rows
## # i 1 more variable: percentage_unresolved <dbl>
```

```r
# Filter the top 12 unresolved MOS types
top_unresolved <- percentage_resolved %>%
  slice_max(order_by = unsolved_count, n = 12) %>%
  arrange(desc(percentage_unresolved))

# Plot the unresolved percentages for the top 12 MOS types
ggplot(top_unresolved, aes(x = reorder(mos, percentage_unresolved), y = percentage_unresolved, fill = pe
  geom_col() +
  geom_text(aes(label = sprintf("%.1f%%", percentage_unresolved)), position = position_stack(vjust = 0.5
  labs(
    x = "MOS Value",
    y = "Percentage Unresolved",
    title = "           Top 12 MOS Values for Account Status 'A' Sorted by Unresolved Percentage"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  coord_flip() +
  scale_fill_gradient(low = "pink", high = "darkred")
```

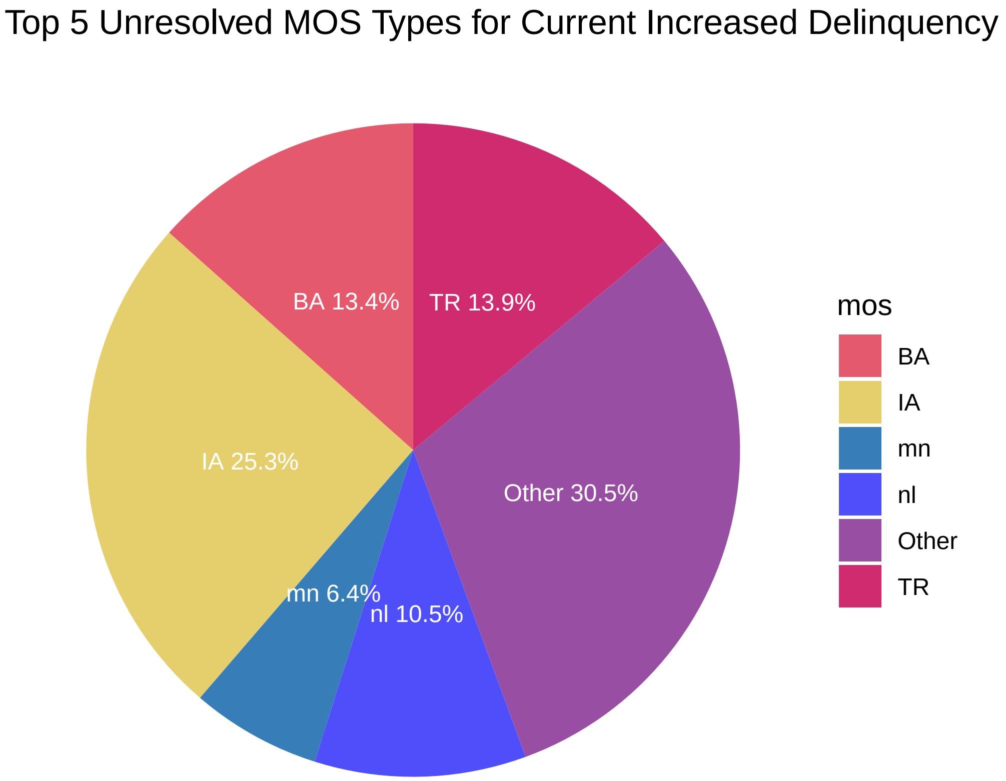


```
# Step 1: Filter the data for account_status_13_march 'A' and calculate counts
filtered_data <- separated_data %>%
  filter(card_activation_status_13_march %in% c(7)) %>%
  count(mos)

# Step 2: Identify the top 5 mos types based on count
top_5_mos <- filtered_data %>%
  top_n(5, wt = n)

# Step 3: Create a dataset with an "Other" category for MOS types outside the top 5
pie_data <- filtered_data %>%
  mutate(mos = if_else(mos %in% top_5_mos$mos, as.character(mos), "Other")) %>%
  group_by(mos) %>%
  summarise(count = sum(n)) %>%
  mutate(percentage = count / sum(count) * 100) %>%
  ungroup()  # Ensure that the data is no longer grouped for plotting

# Define colors for the top 5 MOS types plus "Other", adjust the number of colors accordingly
pie_chart_colors <- c("#E45A6C", "#E4cf6c", "#377EB8", "#4f4FfA", "#984EA3", "#cF2c6f", "#cF2ccf", "#af

# Step 4: Create the pie chart
ggplot(pie_data, aes(x = "", y = percentage, fill = mos)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
  scale_fill_manual(values = pie_chart_colors) +
  theme_void() +
  geom_text(aes(label = paste(mos, sprintf("%.1f%%", percentage))),
            position = position_stack(vjust = 0.5),
            color = "white", size = 3) +
  labs(title = "Top 5 Unresolved MOS Types for Card Activation Status 7") +
  theme(legend.position = "right")
```



```
# Step 1: Filter the data for account_status_13_march 'A' and calculate counts
filtered_data <- separated_data %>%
  filter(delinquency_compare_13_March %in% c("M")) %>%
  count(mos)

# Step 2: Identify the top 5 mos types based on count
top_5_mos <- filtered_data %>%
  top_n(5, wt = n)

# Step 3: Create a dataset with an "Other" category for MOS types outside the top 5
pie_data <- filtered_data %>%
  mutate(mos = if_else(mos %in% top_5_mos$mos, as.character(mos), "Other")) %>%
  group_by(mos) %>%
  summarise(count = sum(n)) %>%
  mutate(percentage = count / sum(count) * 100) %>%
  ungroup()  # Ensure that the data is no longer grouped for plotting

# Define colors for the top 5 MOS types plus "Other", adjust the number of colors accordingly
pie_chart_colors <- c("#E45A6C", "#E4cf6c", "#377EB8", "#4f4FfA", "#984EA3", "#cF2c6f", "#cF2ccf", "#af

# Step 4: Create the pie chart
ggplot(pie_data, aes(x = "", y = percentage, fill = mos)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0) +
```

```r
  scale_fill_manual(values = pie_chart_colors) +
  theme_void() +
  geom_text(aes(label = paste(mos, sprintf("%.1f%%", percentage))),
            position = position_stack(vjust = 0.5),
            color = "white", size = 3) +
  labs(title = "Top 5 Unresolved MOS Types for Current Increased Delinquency") +
  theme(legend.position = "right")
```


Top 5 Unresolved MOS Types for Current Increased Delinquency
BA 13.4%
TR 13.9%
IA 25.3%
Other 30.5%
mn 6.4%
nl 10.5%
mos
BA
IA
mn
nl
Other
TR