Analyze Ranking

Analyzing and Visualizing Ranked Data

Objective

We aim to:

1. Analyze the “Rank your concerns” column, where each respondent has ranked their concerns in order of priority.
2. Break down the rankings, calculate statistics (frequency, average rank, etc.), and visualize the insights.

Steps Overview

1. Read the dataset and dynamically select the “Rank your concerns” column.
2. Split the semicolon-separated rankings into individual concerns and assign ranks based on their position.
3. Aggregate rankings to calculate metrics like frequency, average rank, and rank distribution.
4. Visualize the results to highlight the most and least prioritized concerns.

R Code with Explanations

```
# Load required libraries
library(dplyr)
library(tidyr)
library(ggplot2)

# Step 1: Load the dataset
# Replace 'path_to_file' with the path to your dataset file.
students_data <- read.csv("path_to_file")

# Step 2: Select the "Rank your concerns" column
# Here, replace "Rank.your.concerns" with the actual column name in your dataset if it differs.
concerns_column <- "Rank.your.concerns"

# Step 3: Split and Process the Rankings
# Explanation:
# - Each entry in the "Rank your concerns" column is a semicolon-separated list of concerns ranke
# - We split these rankings into separate rows, assigning a rank value to each concern based on i

concerns_df <- students_data %>%
  select(ID = row_number(), Concerns = !!sym(concerns_column)) %>% # Dynamically select the colum
  separate_rows(Concerns, sep = ";") %>%                          # Split semicolon-separated co
  filter(Concerns != "") %>%                                      # Remove any empty values
  group_by(ID) %>%
  mutate(Rank = row_number()) %>%                                 # Assign rank based on positio
```

```
  ungroup()

# Step 4: Analyze the Rankings
# Explanation:
# - Group by each unique concern to calculate:
#   - Count: How often the concern appears in any rank position.
#   - Avg_Rank: The average rank (lower values mean higher priority).
#   - Min_Rank and Max_Rank: The range of ranks assigned to each concern.
rank_analysis <- concerns_df %>%
  group_by(Concerns) %>%
  summarise(
    Count = n(),                    # Total appearances of each concern
    Avg_Rank = mean(Rank),          # Average rank for each concern
    Min_Rank = min(Rank),           # Minimum rank (highest priority)
    Max_Rank = max(Rank)            # Maximum rank (lowest priority)
  ) %>%
  arrange(Avg_Rank)                 # Sort by average rank (priority)

# Print the summary table
print(rank_analysis)

# Step 5: Visualize the Results
# Explanation:
# - A bar chart to show the frequency of each concern.
# - Concerns are ordered by their average rank to highlight priorities.
ggplot(rank_analysis, aes(x = reorder(Concerns, Avg_Rank), y = Count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(
    title = "Frequency of Concerns",
    x = "Concern",
    y = "Frequency"
  ) +
  coord_flip() # Flip the chart for better readability
```

Analysis and Visualization Rationale

Analyzing Ranking Data

1. **Frequency**: How often each concern appears provides insight into its overall importance.
2. **Average Rank**: A lower average rank indicates higher priority since it means the concern is placed higher in respondents' rankings.
3. **Range (Min/Max Rank)**: Understanding the variability in ranks reveals whether a concern is consistently prioritized or varies across respondents.

Visualization

- **Bar Chart**: Visualizes the frequency of concerns to easily identify which issues are most common.
- **Sorting by Average Rank**: Ensures that higher-priority concerns are emphasized visually.

Expected Insights

1. **Top Priorities**: Concerns with the lowest average rank and highest frequency.
2. **Consistency**: Concerns with a narrow range between min and max rank are consistently prioritized.
3. **Outliers**: Concerns that are rarely ranked high or low could indicate polarizing issues.