To create a **gender distribution table program-wise** from your dataset with the provided column names, I'll guide you step by step. Here's how you can achieve it in R:

**1. Load the Dataset**

Assume your dataset is named required\_details and contains the columns PROGRAMME.NAME and GENDER:

# Check the structure of the dataset

str(phd)

# Ensure the relevant columns exist

# Columns we need: PROGRAMME.NAME and GENDER

if (!("PROGRAMME.NAME" %in% colnames(phd) && "GENDER" %in% colnames(phd))) {

stop("Required columns (PROGRAMME.NAME and GENDER) are missing in the dataset.")

}

**2. Create Gender Distribution Table**

Group the data by PROGRAMME.NAME and GENDER, then calculate the count of students in each group:

# Create the gender distribution table

gender\_distribution <- phd %>%

group\_by(PROGRAMME.NAME, GENDER) %>%

summarise(Count = n(), .groups = "drop") %>%

arrange(PROGRAMME.NAME)

# View the result

print(gender\_distribution)

This will give you a long-format table like:

| **PROGRAMME.NAME** | **GENDER** | **Count** |
| --- | --- | --- |
| Program A | Male | 1200 |
| Program A | Female | 800 |
| Program B | Male | 1000 |
| Program B | Female | 900 |

**3. Pivot the Table (Optional)**

To make the table more intuitive, you can pivot it to a wide format:

gender\_distribution\_wide <- gender\_distribution %>%

pivot\_wider(names\_from = GENDER, values\_from = Count, values\_fill = 0)

# View the wide-format table

print(gender\_distribution\_wide)

This will result in:

| **PROGRAMME.NAME** | **Male** | **Female** |
| --- | --- | --- |
| Program A | 1200 | 800 |
| Program B | 1000 | 900 |

**4. Save the Gender Distribution Table**

If you want to save this table as a CSV file for further analysis:

R

CopyEdit

write.csv(gender\_distribution\_wide, "gender\_distribution\_by\_programme.csv", row.names = FALSE)

**5. Visualize Gender Distribution**

To create a bar chart showing gender distribution for each program:

R

CopyEdit

library(ggplot2)

# Create a bar plot

ggplot(gender\_distribution, aes(x = PROGRAMME.NAME, y = Count, fill = GENDER)) +

geom\_bar(stat = "identity", position = "dodge") +

labs(title = "Gender Distribution Across Programmes",

x = "Programme Name", y = "Number of Students") +

theme(axis.text.x = element\_text(angle = 90, hjust = 1))

**6. Explanation of the Code**

1. **group\_by(PROGRAMME.NAME, GENDER)**: Groups the data by program name and gender.
2. **summarise(Count = n())**: Counts the number of rows for each combination of program and gender.
3. **pivot\_wider()**: Converts the long format table into a wide format, where each gender gets its own column.
4. **ggplot2**: Visualizes the gender distribution as a bar plot.

**7. Ensure Data Quality**

* **Check for Missing or Inconsistent Data:**

R

CopyEdit

# Check for missing values

sum(is.na(required\_details$PROGRAMME.NAME))

sum(is.na(required\_details$GENDER))

# Check unique values in GENDER

unique(required\_details$GENDER)

* If any inconsistencies (e.g., blank or misspelled gender values), clean the data first:

R

CopyEdit

# Replace inconsistent values with NA

required\_details <- required\_details %>%

mutate(GENDER = case\_when(

GENDER %in% c("Male", "Female") ~ GENDER,

TRUE ~ NA\_character\_

))

# Remove rows with missing PROGRAMME.NAME or GENDER

required\_details <- required\_details %>%

filter(!is.na(PROGRAMME.NAME) & !is.na(GENDER))

Let me know if you need further assistance! 😊

To arrange the gender distribution table in **descending order**, you can use the arrange() function from the dplyr package. Here's how to do it step by step:

**1. Arrange the Long-Format Table in Descending Order**

If you want to sort the table by the Count column in descending order:

R

CopyEdit

library(dplyr)

# Arrange the gender\_distribution table in descending order by Count

gender\_distribution\_desc <- gender\_distribution %>%

arrange(desc(Count))

# View the result

print(gender\_distribution\_desc)

This will sort the entire table so that the rows with the highest count appear first.

**2. Arrange the Wide-Format Table in Descending Order**

If you have pivoted the table to the wide format and want to arrange it by the total number of students (Male + Female):

R

CopyEdit

# Add a Total column for sorting

gender\_distribution\_wide <- gender\_distribution\_wide %>%

mutate(Total = Male + Female) %>%

arrange(desc(Total))

# View the result

print(gender\_distribution\_wide)

This will sort the programs in descending order based on the total number of students.

**3. Explanation**

1. **arrange(desc(Count))**: Sorts the long-format table by the Count column in descending order.
2. **mutate(Total = Male + Female)**: Adds a Total column to the wide-format table, which is the sum of Male and Female counts for each program.
3. **arrange(desc(Total))**: Sorts the wide-format table based on the total number of students in descending order.

**Example Output for Wide Format (Descending Order by Total)**

| **PROGRAMME.NAME** | **Male** | **Female** | **Total** |
| --- | --- | --- | --- |
| Program B | 1000 | 900 | 1900 |
| Program A | 1200 | 800 | 2000 |
| Program C | 1100 | 700 | 1800 |

Let me know if you'd like further clarification! 😊