Analyzing Data Distribution using Frequency and Contingency Tables

Table of contents

0.1 The Dataset: Course Evaluation

Course Evaluation

0.2 Introduction to the Dataset

This dataset represents student data, comprising of fields such as majors, courses, scores, grades, and evaluations. The data provides insights into students' academic performance and feedback across different disciplines.

Column	Description
`Major`	The major or department of the course (e.g., `CS` for Computer Science).
`Course`	The specific course identifier within the major (e.g., `CS_101`).
`Score`	Numerical score obtained by a student in the course.
`Grade`	Alphabetical grade awarded to the student based on the score.
`Evaluation`	A numerical rating (1-10 scale) representing student evaluations for the course.

0.3 Dataset Loading

```
df <- read.csv("https://raw.githubusercontent.com/ahmedmoustafa/datasets/main/evaluation/e
head(df)
```

Major	Course	Score	Grade	Evaluation
CS	CS_101	46.10	F	4
CS	CS_102	81.71	B	8
CS	CS_103	68.13	D	9
CS	CS_101	62.55	D	6
CS	CS_102	76.94	C	7
CS	CS_103	72.96	C	8

0.4 Frequency Tables

A frequency table is a way to organize data by recording the number of times each value or range of values appears in the dataset. The formula for frequency for a category i is given by:

$$f_i = \text{ Number of times category } i \text{ appears in the data}$$

Example: Suppose we have the following grades: 85, 90, 88, 82, 92, and we use two bins: 80-89 and 90-99. The frequency table would look like:

Bin	Frequency
80-89	3
90-99	2

0.5 Frequency Metrics

- **Normalized Frequency**

$$f_{norm} = \frac{\text{Frequency of bin}}{\text{Total number of data points}}$$

- **Cumulative Frequency**

$$f_{cum} = f_1 + f_2 + \cdots + f_i$$

- **Normalized Cumulative Frequency**

$$f_{cum\_norm} = \frac{f_1 + f_2 + \cdots + f_i}{n}$$

Example:

Using the same data from the previous slide:

Bin	f	f_{norm}	f_{cum}	f_{cum_norm}
80-89	3	0.6	3	0.6
90-99	2	0.4	5	1

0.6 For Qualitative Data

Qualitative (or categorical) data can be summarized using basic frequency tables. Each unique category gets its own entry in the table.

```r
table(df$Major) # Creating a frequency table for the 'Major' column
```

BA	CS	ME
600	450	300

0.7 For Quantitative Data

Quantitative data requires binning, where data is grouped into ranges. There are several methods to decide on the number of bins:

1. **Square-root Rule**: number of bins $= \sqrt{n}$
2. **Sturges' Rule**: number of bins $= 1 + log_2(n)$
3. **Rice Rule**: number of bins $= 2 \times \sqrt[3]{n}$
4. **Freedman-Diaconis Rule**:
   - bin width $= 2 \times \frac{\text{IQR}(x)}{\sqrt[3]{n}}$
   - number of bins $= \left\lceil \frac{\max(x) - \min(x)}{\text{bin width}} \right\rceil$

0.8 Number of Bins

Let's calculate the number of bins using these methods, for $n = 1350$:

```
# Number of data points
n = length(df$Score)

# Calculating number of bins using various methods
bins_sqrt = round(sqrt(n)) # Square-root
bins_sturges = round(log2(n) + 1) # Sturges
bins_rice = round(2 * (n^(1/3))) # Rice
iqr = IQR(df$Score) # IQQ
bin_width_fd = 2 * iqr / (n^(1/3)) # Bin width for Freedman-Diaconis
bins_fd = round((max(df$Score) - min(df$Score)) / bin_width_fd) # Freedman-Diaconis

data.frame(Method=c("Square root", "Sturges", "Rice", "Freedman-Diaconis"), Number_of_Bins
```

Method	Number_of_Bins
Square root	37
Sturges	11
Rice	22
Freedman-Diaconis	33

0.9 Creating the Frequency Table

We'll calculate frequencies using Sturges' rule for the sake of demonstration:

```
# Break points for the bins using square root rule
break_points = seq(min(df$Score),
                   max(df$Score),
                   length.out=bins_sturges+1)
break_points
```

```
 [1]  17.21000  24.73636  32.26273  39.78909  47.31545  54.84182  62.36818
 [8]  69.89455  77.42091  84.94727  92.47364 100.00000
```

```
# Calculate frequencies
frequencies = table(cut(df$Score,
                        breaks=break_points,
                        include.lowest=TRUE))
frequencies
```

[17.2,24.7]	(24.7,32.3]	(32.3,39.8]	(39.8,47.3]	(47.3,54.8]	(54.8,62.4]	(62.4,69.9]	(69.9,77.4]	(77.4,84.9]	(84.9,92.5]	(92.5,100]
1	0	5	19	72	155	293	401	271	106	27

```r
# Calculating normalized and cumulative frequencies
norm_freq <- frequencies / sum(frequencies)
cum_freq <- cumsum(frequencies)
norm_cum_freq <- cumsum(norm_freq)

data.frame(Bin = names(frequencies),
           Frequency = as.vector(frequencies),
           Normalized_Frequency = as.vector(norm_freq),
           Cumulative_Frequency = as.vector(cum_freq),
           Normalized_Cumulative_Frequency = as.vector(norm_cum_freq))
```

Bin	Frequency	Normalized_Frequency	Cumulative_Frequency	Normalized_Cumulative_Frequency
[17.2,24.7]	1	0.0007407	1	0.0007407
(24.7,32.3]	0	0.0000000	1	0.0007407
(32.3,39.8]	5	0.0037037	6	0.0044444
(39.8,47.3]	19	0.0140741	25	0.0185185
(47.3,54.8]	72	0.0533333	97	0.0718519
(54.8,62.4]	155	0.1148148	252	0.1866667
(62.4,69.9]	293	0.2170370	545	0.4037037
(69.9,77.4]	401	0.2970370	946	0.7007407
(77.4,84.9]	271	0.2007407	1217	0.9014815
(84.9,92.5]	106	0.0785185	1323	0.9800000
(92.5,100]	27	0.0200000	1350	1.0000000

0.10 Using `DescTools::Freq()`

```r
DescTools::Freq(df$Score) # Creating a frequency table for the 'Score' column using DescTo
```

level	freq	perc	cumfreq	cumperc
[15,20]	1	0.0007407	1	0.0007407
(20,25]	0	0.0000000	1	0.0007407
(25,30]	0	0.0000000	1	0.0007407
(30,35]	0	0.0000000	1	0.0007407
(35,40]	5	0.0037037	6	0.0044444

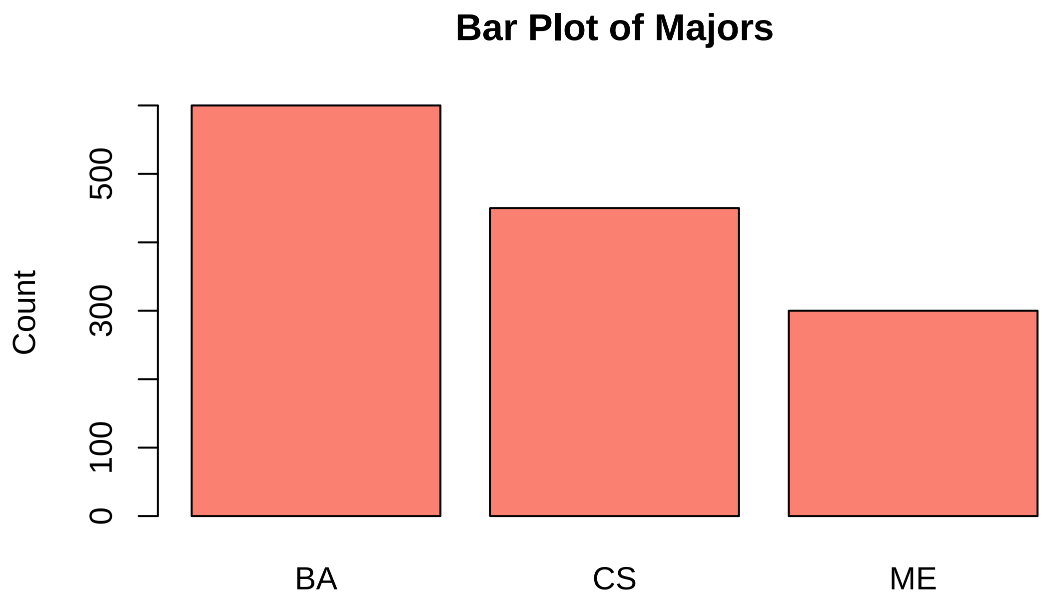
level	freq	perc	cumfreq	cumperc
(40,45]	9	0.0066667	15	0.0111111
(45,50]	29	0.0214815	44	0.0325926
(50,55]	54	0.0400000	98	0.0725926
(55,60]	84	0.0622222	182	0.1348148
(60,65]	169	0.1251852	351	0.2600000
(65,70]	202	0.1496296	553	0.4096296
(70,75]	263	0.1948148	816	0.6044444
(75,80]	247	0.1829630	1063	0.7874074
(80,85]	156	0.1155556	1219	0.9029630
(85,90]	99	0.0733333	1318	0.9762963
(90,95]	18	0.0133333	1336	0.9896296
(95,100]	14	0.0103704	1350	1.0000000

0.11 Visualization of Frequency Tables

Bar plots and histograms and provide visual representations of frequency tables:

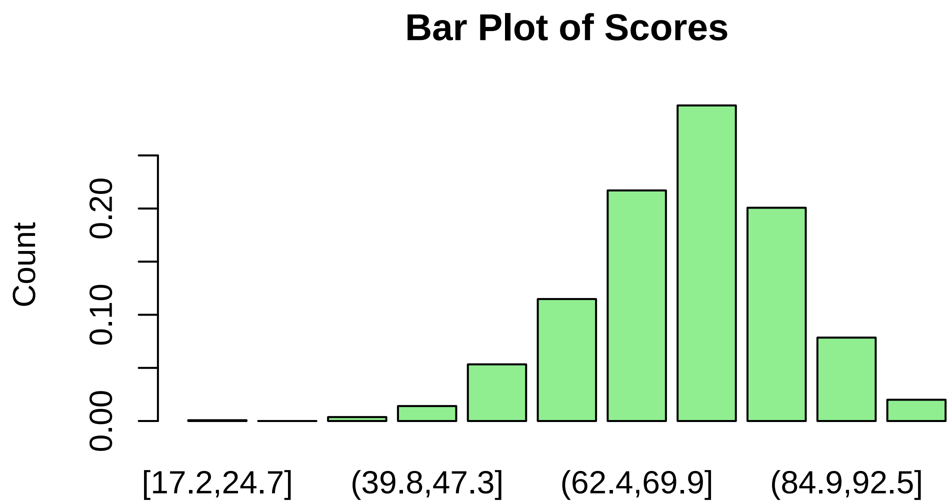
0.12 Qualitative using `barplot()`

```
# Bar plot for 'Major'
barplot(table(df$Major), col="salmon", border="black", main="Bar Plot of Majors", ylab="Co
```



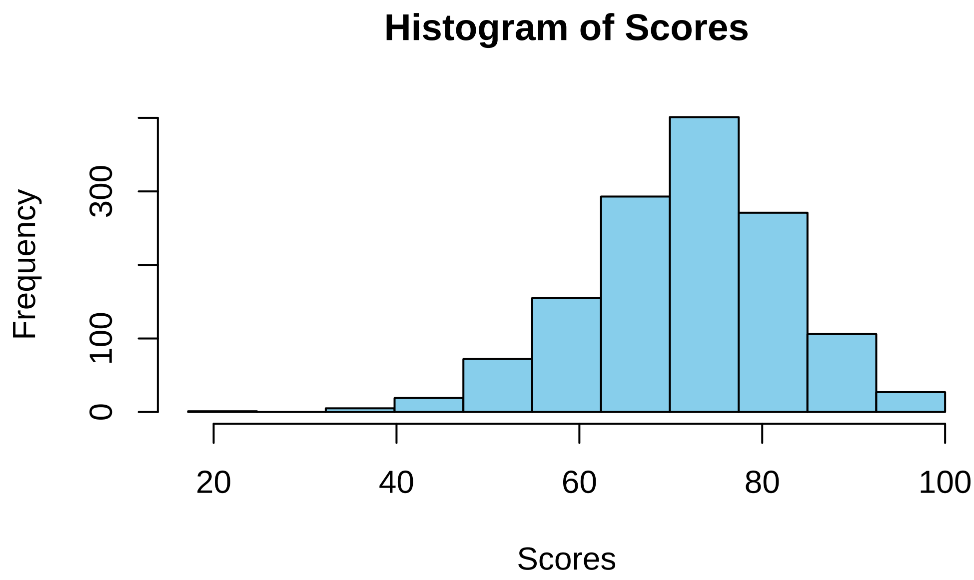
0.13 Quantitative using `barplot()`

```
# Bar plot for 'Major'
barplot(norm_freq, col="lightgreen", border="black", main="Bar Plot of Scores", ylab="Coun
```



0.14 Quantitative using `hist()`

```
# Histogram for 'Score'
hist(df$Score, breaks=break_points, col="skyblue", border="black", main="Histogram of Scor
```



0.15 Contingency Tables: Multivariate Categorical Data

Contingency tables help to understand the relationship between **two categorical** variables by listing the frequency of every combination of categories:

$$f_{ij} = \text{Number of occurrences where variable 1 is in category } i \text{ and variable 2 is in category } j$$

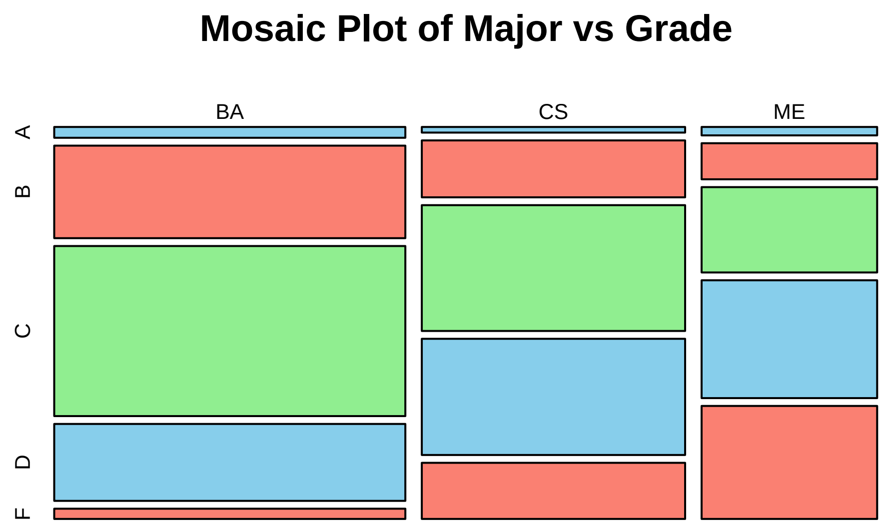
```r
# Creating a contingency table for 'Major' and 'Grade'
contingency <- table(df$Major, df$Grade)
contingency
```

/	A	B	C	D	F
BA	18	154	283	128	17
CS	7	71	157	145	70
ME	7	30	71	98	94

0.16 Visualization of Contingency Tables

Mosaic plots provide a visual representation of contingency tables, highlighting the distribution and relationship between two categorical variables.

```
# Mosaic plot for the contingency table
mosaicplot(contingency, main="Mosaic Plot of Major vs Grade", color=c("skyblue", "salmon",
```



0.17 Exercise

1. Frequency Table for the `Evaluation` Column

a. Construct a frequency table for the `Evaluation` column in the provided dataset.
b. Visualize the frequency table using an appropriate plot.
c. Analyze the resulting visualization and articulate any relationships, trends, or patterns observed in the `Evaluation` data.

2. Relationship between `Score` and `Evaluation`

a. Employ suitable visualization techniques to explore the relationship between the `Score` and `Evaluation` columns in the dataset.
b. Examine the visual representation and infer any relationships, trends, or patterns between `Score` and `Evaluation`.

3. Contingency Table for Computer Science (`CS`) Major Courses

a. Filter the dataset to include only rows where the courses belong to the Computer Science major.
b. Develop a contingency table between `Courses` and `Grades` from the filtered data.
c. Visualize the contingency table using suitable graphical representations.
d. Analyze the visualization and deduce any notable relationships, trends, or patterns between different courses and grades within the Computer Science major.

0.18 In Summary

- Frequency tables are fundamental in data analysis to understand data distribution.
- The choice of binning is pivotal for meaningful interpretation of quantitative data.
- Contingency tables offer insights into the relationships between two categorical variables.
- Both tables can be visualized effectively for better understanding and interpretation.