Categorical Data (3 of 3)

July 28, 2023 V1.1 (Work in progress)

Overview of Bivariate and Multivariate Categorical Variables

1. Bivariate and multivariate categorical variables allow us to analyze and comprehend relationships between two or more categorical variables respectively.
2. Bivariate analysis involves examining the relationship between two variables. For instance, we might examine the relationship between a person's gender (male, female, or non-binary) and whether they own a car (yes or no). By exploring these two categorical variables together, we can discern potential correlations or associations.
3. On the other hand, multivariate analysis involves the simultaneous observation and analysis of more than two variables. This form of analysis can help reveal complex interactions and dependencies between multiple variables that cannot be detected in bivariate analyses. For example, we might want to explore the relationship between a person's gender, car ownership status, and their level of education (high school, bachelor's, master's, etc.). Both bivariate and multivariate analyses are essential in statistical and data analysis as they allow us to uncover relationships and patterns in data

Bivariate Categorical Variables

1. Bivariate categorical data can be understood as a type of data where we are examining two categorical variables simultaneously, where we need to explore relationships or differences between the two factors, using contingency tables.
2. **Contingency Table**: A contingency table, also known as a cross-tabulation or crosstab, is a type of table in a matrix format that displays the (multivariate) frequency distribution of the variables. In the case of a univariate factor variable, a contingency table is essentially the same as a frequency table, as there's only one variable involved. In more complex analyses involving two or more variables, contingency tables provide a way to examine the interactions between the variables [(Agresti, 2007). [1]

3. Data: Let us work with the same mtcars data from the previous chapter. Suppose we have run the following code:

```
# Load the required libraries, suppressing annoying startup messages
library(tibble)
suppressPackageStartupMessages(library(dplyr))
# Read the mtcars dataset into a tibble called tb
data(mtcars)
tb <- as_tibble(mtcars)
attach(tb)
# Convert several numeric columns into factor variables
tb$cyl <- as.factor(tb$cyl)
tb$vs <- as.factor(tb$vs)
tb$am <- as.factor(tb$am)
tb$gear <- as.factor(tb$gear)
```

Multivariate Categorical Variables

Three Way Relationships

1. When exploring the intricate relationships within multivariate categorical data, we often extend our analysis from two-way relationships to three-way relationships. While the former delves into associations between two categorical variables, the latter deepens our understanding by considering interactions among three categorical variables concurrently.
2. Graphically, three-way relationships can be represented in various forms, such as three-dimensional contingency tables, side-by-side mosaic plots, or even three-dimensional bar plots. However, it's crucial to note that these visual representations can become complicated and challenging to decipher as the number of categories within each variable rises (Agresti, 2002).
3. When dealing with a three-way relationship, our focus is on three categorical variables and how they interact with each other. Such interactions can be manifest in the form of changes in the relationship between two variables based on the values of the third variable. Alternatively, we might seek to comprehend how all three variables collectively influence the observed data.
4. The R language, versatile as it is, provides multiple functions for creating contingency tables for multivariate categorical data. In this case, we're focusing on the table(), xtabs(), and ftable() functions for forming a three-way contingency table (R Core Team, 2020). Here are the given code lines:

5. table(): We can create a three-way contingency table of cyl, gear, and am is created using the table() function.

```
table(cyl,
      gear,
      am)
```

```
, , am = 0

   gear
cyl  3  4  5
  4  1  2  0
  6  2  2  0
  8 12  0  0

, , am = 1

   gear
cyl  3  4  5
  4  0  6  2
  6  0  2  1
  8  0  0  2
```

- When we run this code, the output is a three-dimensional contingency table showing the frequencies of all combinations of the three variables. Each cell in the resulting table represents the number of observations for a particular combination of cyl, gear, and am categories.
- Notice that we are segmenting the tables based on the 3rd argument given the table function, which is the transmission am.

We could alternately run the following code and instead segment the tables based on the number of cylinders cyl.

```
table(am,
      gear,
      cyl)
```

```
, , cyl = 4

   gear
```

```
am   3  4  5
  0  1  2  0
  1  0  6  2

, , cyl = 6

   gear
am   3  4  5
  0  2  2  0
  1  0  2  1

, , cyl = 8

   gear
am   3  4  5
  0 12  0  0
  1  0  0  2
```

2. xtabs(): We can also create a three-way contingency table of cyl, gear, and am using the xtabs() function

```
xtabs(~ cyl + gear + am
      , data = tb)
```

```
, , am = 0

   gear
cyl  3  4  5
  4  1  2  0
  6  2  2  0
  8 12  0  0

, , am = 1

   gear
cyl  3  4  5
  4  0  6  2
  6  0  2  1
  8  0  0  2
```

- Here, the formula ~ cyl + gear + am defines the three variables we are interested in.

3. ftable(): The ftable() function is employed to generate a 'flat' contingency table, which is essentially a higher-dimensional contingency table displayed in a two-dimensional format (R Core Team, 2020). We can also create a three-way contingency table of gear, cyl, and am using the following code:

```
ftable(gear + cyl ~ am,
       data = tb)
```

```
   gear  3        4        5
   cyl   4  6  8  4  6  8  4  6  8
am
0        1  2 12  2  2  0  0  0  0
1        0  0  0  6  2  0  2  1  2
```

In this code, ftable(gear + cyl ~ am, data = tb), we are asking R to arrange the gear and cyl variables in the rows and the am variable in the columns.

The ~ operator separates the variables that will be displayed in rows (on the left) and columns (on the right) in the resulting table.

The + operator denotes that both gear and cyl will be included in the row labels.

```
ftable(gear ~ cyl + am,
       data = tb)
```

```
       gear  3  4  5
cyl am
4   0        1  2  0
    1        0  6  2
6   0        2  2  0
    1        0  2  1
8   0       12  0  0
    1        0  0  2
```

- This variation of code, ftable(gear ~ cyl + am, data = tb), it is structured slightly differently.
- Here, the gear variable forms the row and both cyl and am variables form the columns of the flat contingency table.

- In both scenarios, an ftable provides a more compact view of the three-way relationship among the gear, cyl, and am variables. However, the orientation of the variables in the rows and columns changes, providing different views of the data and potentially making certain patterns more evident depending on the question we're trying to answer.
- The exact choice between ftable(gear + cyl ~ am, data = tb) and ftable(gear ~ cyl + am, data = tb) will depend on what specific relationships you're most interested in exploring in your data.

Four Way Relationships

```
ftable(am + cyl ~ gear + vs,
       data = tb)
```

```
        am   0        1
        cyl  4  6  8  4  6  8
gear vs
3    0       0  0 12  0  0  0
     1       1  2  0  0  0  0
4    0       0  0  0  0  2  0
     1       2  2  0  6  0  0
5    0       0  0  0  1  1  2
     1       0  0  0  1  0  0
```

```
ftable(am + cyl + vs ~ gear,
       data = tb)
```

```
     am   0                 1
     cyl  4     6     8     4     6     8
     vs   0  1  0  1  0  1  0  1  0  1  0  1
gear
3         0  1  0  2 12  0  0  0  0  0  0  0
4         0  2  0  2  0  0  0  6  2  0  0  0
5         0  0  0  0  0  0  1  1  1  0  2  0
```

In this example, we establish a four-way contingency table containing am, cyl, gear, and vs using the ftable() function. The frequency of each grouping of categories is displayed in the table that results. There are two vehicles with a 6-cylinder, 3-gear, automatic transmission, and inline engine, for instance, and three vehicles with four cylinders. The resulting table, which has two two-dimensional tables for each level of the am variable, is four-dimensional.

Visualization of Multivariate Categorical Variables

```
# Load the mtcars dataset
data(mtcars)

# Install and load the vcd package (if it's not already installed)
library(vcd)
```

```
Loading required package: grid
```

```
# Create a mosaic plot of mpg (miles per gallon) vs. vs (engine shape)
#mosaic(~ cyl + vs + gear, data = mtcars, main = "Mosaic Plot of MPG vs. VS")
```

References

[1] Agresti, A. (2018). An Introduction to Categorical Data Analysis (3rd ed.). Wiley.

Kabacoff, R. I. (2015). R in Action: Data analysis and graphics with R (2nd ed.). Manning Publications.

Wickham, H., & Grolemund, G. (2016). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media.

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2018). Multivariate data analysis (8th ed.). Cengage Learning.

[2] Unwin, A. (2015). Graphical data analysis with R. CRC Press.

Friendly, M. (2000). Visualizing Categorical Data. SAS Institute.

Hartigan, J. A., & Kleiner, B. (1981). Mosaics for contingency tables. In Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface (pp. 268-273).

[3] Healy, K., & Lenard, M. T. (2014). A practical guide to creating better looking plots in R. University of Oregon. https://escholarship.org/uc/item/07m6r

[4] Meyer, D., Zeileis, A., & Hornik, K. (2020). vcd: Visualizing Categorical Data. R package version 1.4-8. https://CRAN.R-project.org/package=vcd

Friendly, M. (1994). Mosaic displays for multi-way contingency tables. Journal of the American Statistical Association, 89(425), 190-200.

Agresti, A. (2018). An Introduction to Categorical Data Analysis (3rd ed.). Wiley.

[5] R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.