Categorical Data (3 of 3)

July 30, 2023

Overview of Multivariate Categorical Variables

1. Multivariate categorical variables allow us to analyze and comprehend relationships between **three or more** categorical variables respectively.

2. 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

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)
```

Three Way Relationships

1. A Three-Dimensional Contingency Table, often referred to as a 3-way contingency table, is a statistical tool that helps analyze the relationship between three categorical variables. It builds upon the concept of a standard two-dimensional contingency table, which shows the distribution of two categorical variables, by adding a third dimension to the analysis.
2. Imagine a grid-like structure with three axes representing the three variables. The rows correspond to the categories of the first variable, the columns represent the categories of the second variable, and the layers (sheets) represent the categories of the third variable. Each cell within the table contains the frequency or count of observations that belong to a specific combination of the three variables.
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. 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).

Three-Dimensional Contingency Tables

1. 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 is some code:
2. We can create a three-way contingency table of `cyl`, `gear`, and `am` 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`.

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

```r
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
```

4. `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.

5. `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.

```r
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.

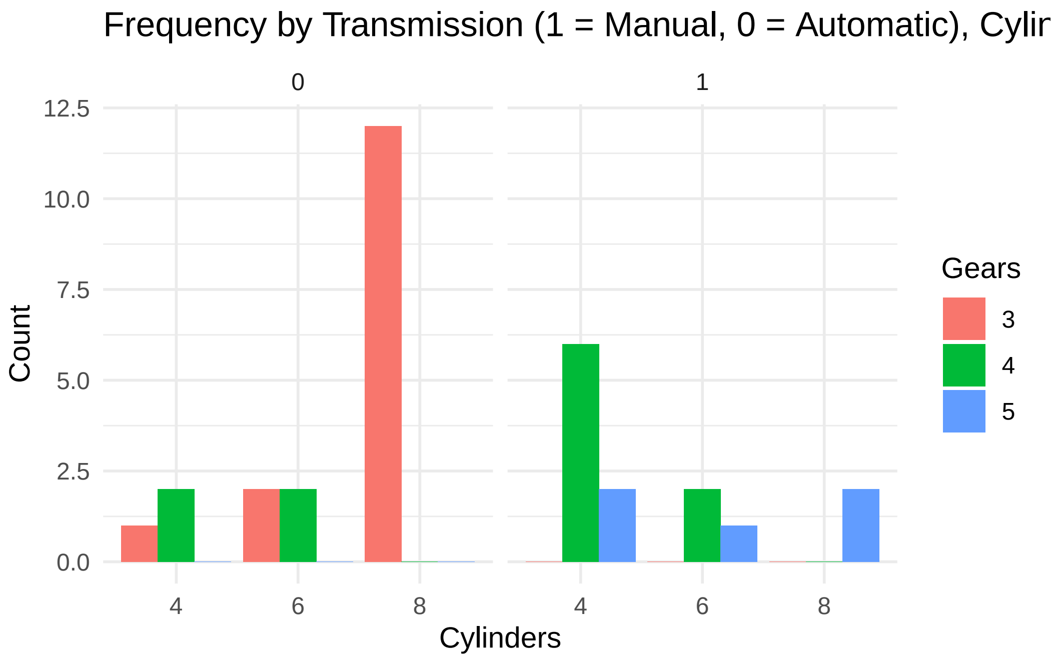
Visualization using a Faceted Bar Plot in `ggplot`

1. A Three-Dimensional Bar Plot is a generalization of a conventional two-dimensional bar graph, expanded into a third dimension. Rather than using bars at specific x coordinates in a two-dimensional plane, we utilize a grid of bars on the x-y plane, extending upwards in the z direction to indicate the data's magnitude. [6]
2. Here is some sample code:

```r
# Load necessary package
library(ggplot2)
```

```
Attaching package: 'ggplot2'

The following object is masked from 'tb':

    mpg
```

```r
# Create a table with count of each combination
count_df <- table(tb$cyl, tb$am, tb$gear)

# Convert table to a data frame for plotting
count_df <- as.data.frame.table(count_df)

# Rename columns
names(count_df) <- c("cyl", "am", "gear", "count")

# Create the plot
ggplot(count_df, aes(x=cyl, y=count, fill=gear)) +
  geom_bar(stat="identity", position="dodge") +
  facet_grid(~am) +
  labs(
    title = "Frequency by Transmission (1 = Manual, 0 = Automatic), Cylinders and Gears",
    x = "Cylinders",
    y = "Count",
    fill = "Gears"
  ) +
  theme_minimal()
```

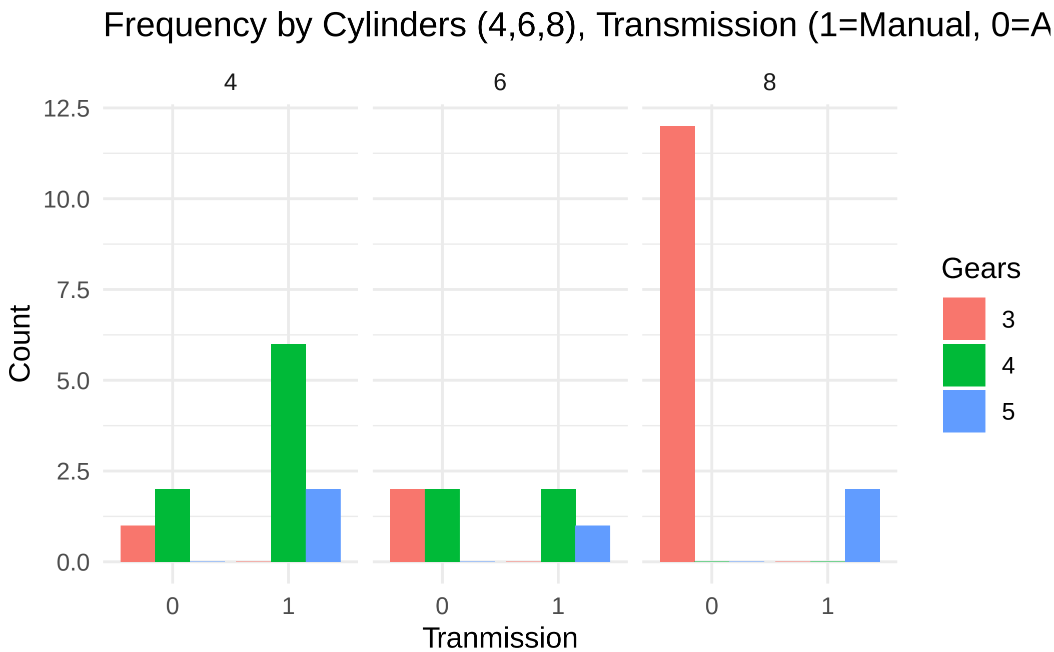


- This code creates a faceted bar plot to visually represent the frequency of combinations of three categorical variables: `cyl` (Cylinders), `am` (Transmission), and `gear` (Gears).
- The line `count_df <- table(tb$cyl, tb$am, tb$gear)` generates a contingency table of the frequencies at each level of the three categorical variables (`cyl`, `am`, `gear`), using the `table()` function.
- Next, `count_df <- as.data.frame.table(count_df)` is used to convert the generated contingency table into a data frame, which can be more conveniently manipulated and visualized using `ggplot2` (Wickham, 2016).
- The names of the data frame's columns are then reassigned using `names(count_df) <- c("cyl", "am", "gear", "count")`. The 'count' column represents the frequency of each combination of the levels of the `cyl`, `am`, and `gear` variables.
- The plot is created using the `ggplot()` function, which initializes a ggplot object. The aesthetic mapping `aes(x=cyl, y=count, fill=gear)` specifies that the x-axis represents `cyl`, the y-axis represents `count`, and the color fill of the bars is based on `gear`.
- The `geom_bar(stat="identity", position="dodge")` function call adds a layer to the plot that depicts the data as a bar chart. The argument `stat="identity"` informs ggplot that the heights of the bars are given in the data (i.e., in the `count` variable), and `position="dodge"` causes bars associated with different levels of `gear` to be drawn side-by-side.
- The `facet_grid(~am)` function call adds facets to the plot based on the `am` variable, creating a separate subplot for each level of `am`.

- The `labs()` function call specifies the labels for the plot, including the title and the x-, y-, and fill-axis labels. The `theme_minimal()` call is used to apply a minimalist aesthetic theme to the plot.
- This code thus provides a clear and insightful visualization of the frequency of each combination of `cyl`, `am`, and `gear`. [7]

3. Here is some alternate sample code:

```
# Load necessary package
library(ggplot2)

# Create a table with count of each combination
count_df <- table(tb$cyl, tb$am, tb$gear)

# Convert table to a data frame for plotting
count_df <- as.data.frame.table(count_df)

# Rename columns
names(count_df) <- c("cyl", "am", "gear", "count")

# Create the plot
ggplot(count_df, aes(x=am, y=count, fill=gear)) +
  geom_bar(stat="identity", position="dodge") +
  facet_grid(~cyl) +
  labs(
    title = "Frequency by Cylinders (4,6,8), Transmission (1=Manual, 0=Automatic) and Gear
    x = "Tranmission",
    y = "Count",
    fill = "Gears"
  ) +
  theme_minimal()
```



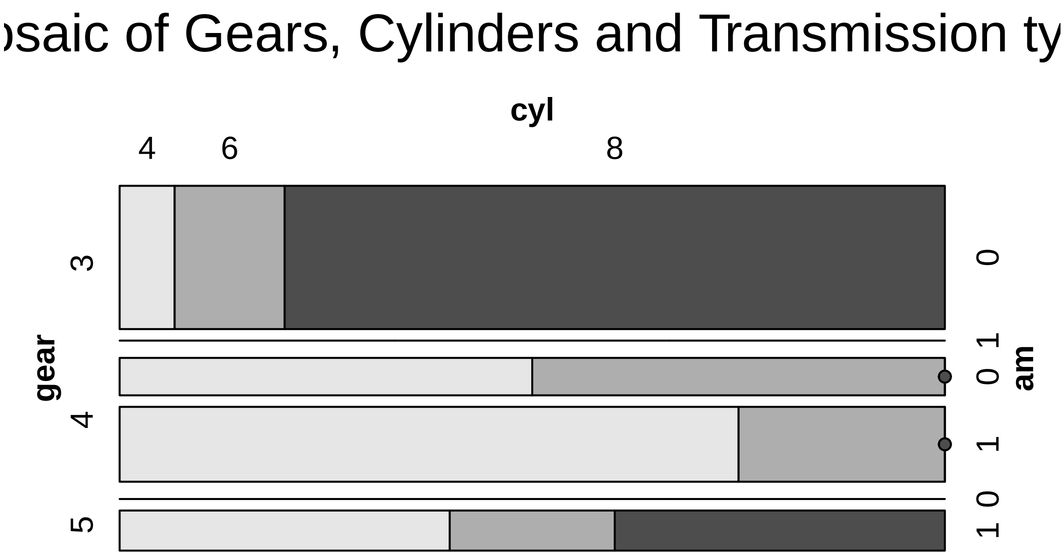
- This code is very similar to the previous one; both create a faceted bar plot to display the frequencies of the levels of three categorical variables from the tb dataset. The main difference between the two lies in the aesthetic mappings and the facet specification in the ggplot() function call.
- In this new code, the x-axis mapping in aes() is changed from cyl (Cylinders) to am (Transmission). Hence, the x-axis of the bar plot will now depict the Transmission type instead of Cylinders.
- Similarly, the facet_grid() function, which was previously applied to am, is now applied to cyl. This means that the plot will now be faceted by the Cylinders variable. Each facet (or subplot) will correspond to a different number of Cylinders (4, 6, or 8). [7]

Visualization using a Mosaic Plot

```
# Load necessary packages
library(vcd)
```

```
Loading required package: grid
```

```
# Create a mosaic plot
vcd::mosaic(~gear+cyl+am,
       data = tb,
       main = "Mosaic of Gears, Cylinders and Transmission type",
       shade = TRUE,
       highlighting = "cyl" )
```



- The provided R code generates a mosaic plot using the `vcd` package, specifically focusing on the `gear`, `cyl`, and `am` variables from the `tb` dataset. A mosaic plot is a visual representation of the frequencies or proportions of combinations of categorical variables.
- `vcd::mosaic(~gear+cyl+am, data = tb, main = "Mosaic of Gears, Cylinders and Transmission type", shade = TRUE, highlighting = "cyl" )` generates the mosaic plot. The `~gear+cyl+am` formula indicates that the mosaic plot should visualize the `gear`, `cyl`, and `am` variables.
- `data = tb` specifies the dataset to be used, which is `tb` in this case.
- `main = "Mosaic of Gears, Cylinders and Transmission type"` sets the main title of the plot.
- `shade = TRUE` means that shading is applied to the cells in the plot. The shading can help to differentiate the cells visually based on the residuals from a model of independence (Meyer, Zeileis, & Hornik, 2006).
- `highlighting = "cyl"` means that the `cyl` variable's levels will be distinctly colored. This highlighting helps to visually emphasize the differences among `cyl` categories in the plot.

The generated mosaic plot provides an effective visual exploration of the joint distribution of `gear`, `cyl`, and `am` variables in the `tb` dataset, highlighting the `cyl` variable.

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
```

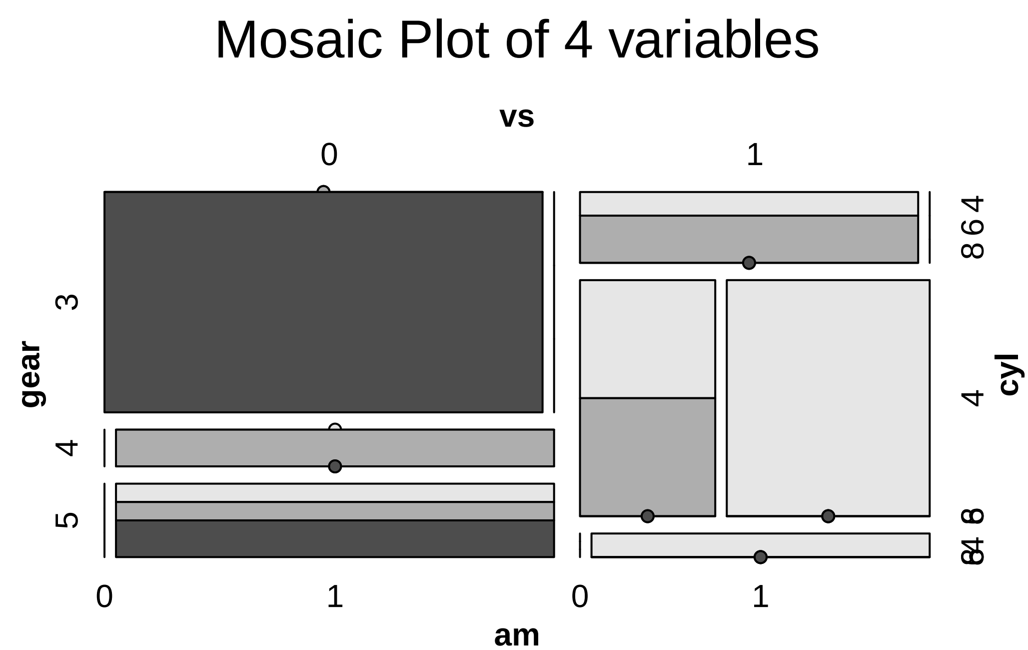
```
ftable(am + cyl + vs ~ gear,
       data = tb)
```

```
     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 code, we establish a four-way contingency table containing `am`, `cyl`, `gear`, and `vs` using the `ftable()` function.

Visualization of Multivariate Categorical Variables

```r
# Create a mosaic plot of mpg (miles per gallon) vs. vs (engine shape)
vcd::mosaic(~ cyl + vs + gear + am,
            data = tb,
            main = "Mosaic Plot of 4 variables",
            shade = TRUE,
            highlighting = "cyl" )
```



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/.

[6]

Unwin, A., Theus, M., & Hofmann, H. (2006). Graphics of large datasets: visualizing a million. Springer Science & Business Media.

[7]

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag.

Meyer, D., Zeileis, A., & Hornik, K. (2006). The strucplot framework: Visualizing multi-way contingency tables with vcd. Journal of Statistical Software, 17(3), 1-48.