Continuous Data (4 of 6)

Aug 4, 2023.

1. THIS CHAPTER explores Continuous x Categorical data using **ggplot2**. Specifically, it demonstrates the use of the popular **ggplot2** package to further explore *bivariate continuous data across categories.*

2. **Data**: Let us work with the same **mtcars** data from the previous chapter. Suppose we run the following code to prepare the data for subsequent analysis. The data is now in a tibble called `tb`:

```
# 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)
# 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)
# Directly access the data columns of tb, without tb$mpg
attach(tb)
```

Summarizing Continuous Data across one Category, using ggplot2

1. We demonstrate the bivariate relationship between Miles Per Gallon (`mpg`) and Cylinders (`cyl`) using `ggplot2`.

```
library(dplyr)

tb %>%
  group_by(cyl) %>%
  summarise(Mean_mpg = mean(mpg, na.rm = TRUE),
            SD_mpg = sd(mpg, na.rm = TRUE))
```

```
# A tibble: 3 x 3
  cyl   Mean_mpg SD_mpg
  <fct>    <dbl>  <dbl>
1 4         26.7   4.51
2 6         19.7   1.45
3 8         15.1   2.56
```

2. Discussion:

- In this code, we use the pipe operator `%\>%` to perform a series of operations. We first group the data by the `cyl` column using the `group_by()` function. We then use `summarise()` to apply the `mean()` and `sd()` functions to the `mpg` column.
- The results are stored in new columns, aptly named `Mean_mpg` and `SD_mpg`.
- We set `na.rm = TRUE` in both `mean()` and `sd()` function calls, to remove any missing values before calculation. [1]

3. We extend this code to demonstrate how to measure the bivariate relationships between multiple continuous variables from the mtcars data and the categorical variable number of Cylinders (`cyl`), using `ggplot2`. Specifically, we consider the continuous variables (i) Miles Per Gallon (`mpg`); (ii) Weight (`wt`); (iii) Horsepower (`hp`) across the number of Cylinders (`cyl`).

```
library(dplyr)
tb %>%
  group_by(cyl) %>%
  summarise(
    Mean_mpg = mean(mpg, na.rm = TRUE),
    SD_mpg = sd(mpg, na.rm = TRUE),
    Mean_wt = mean(wt, na.rm = TRUE),
    SD_wt = sd(wt, na.rm = TRUE),
    Mean_hp = mean(hp, na.rm = TRUE),
    SD_hp = sd(hp, na.rm = TRUE)
    )
```

```
# A tibble: 3 x 7
  cyl   Mean_mpg SD_mpg Mean_wt SD_wt Mean_hp SD_hp
  <fct>    <dbl>  <dbl>   <dbl> <dbl>   <dbl> <dbl>
```

```
1 4          26.7   4.51    2.29 0.570    82.6  20.9
2 6          19.7   1.45    3.12 0.356   122.   24.3
3 8          15.1   2.56    4.00 0.759   209.   51.0
```

4. Discussion:

- With `tb %>%`, we indicate that we are going to perform a series of operations on the `tb` data frame. The next operation is `group_by(cyl)`, which groups the data by the `cyl` variable.
- The `summarise()` function is then used to create a new data frame that summarizes the grouped data. Inside `summarise()`, we calculate the mean and standard deviation (SD) of three variables (`mpg`, `wt`, and `hp`). The`na.rm = TRUE` argument inside `mean()` and `sd()` functions is used to exclude any NA values from these calculations.
- The resulting calculations are assigned to new variables (`Mean_mpg`, `SD_mpg`, `Mean_wt`, `SD_wt`, `Mean_hp`, and `SD_hp`) which will be the columns in the summarised data frame. The summarised data will contain one row for each group (in this case, each unique value of `cyl`), and columns for each of the summary statistics.
- To summarize, this script groups the data in the `tb` tibble by `cyl` and then calculates the mean and standard deviation of the `mpg`, `wt`, and `hp` variables for each group. [1]

Visualizing Continuous Data across one Category, using ggplot2

Let's take a closer look at some of the most effective ways of visualizing continuous data, across one Category, **using ggplot2**, including

(i) Histograms, using ggplot2;

(ii) PDF and CDF Density plots, using ggplot2;

(iii) Box plots, using ggplot2;

(iv) Bee Swarm plots, using ggplot2;

(v) Violin plots, using ggplot2;

(vi) Q-Q plots, using ggplot2.

Summarizing Continuous Data across two Categories using ggplot2

Visualizing Continuous Data across two Categories using ggplot2

References

[1]

Wickham, H., François, R., Henry, L., & Müller, K. (2021). dplyr: A Grammar of Data Manipulation. R package version 1.0.7. https://CRAN.R-project.org/package=dplyr