Descriptive Statistics using R

Table of contents

1 The Dataset: Student Transcript

Student Transcript

2 Objective

Analyzing a student's transcript dataset to understand performance metrics.

3 Dataset Overview

- Columns: year, semester, course_number, credits, letter_grade, numerical_value (GPA)
- Four academic years of data
- Grades for both major and non-major courses

4 Loading the Dataset

```
df = read.csv ("https://raw.githubusercontent.com/ahmedmoustafa/datasets/main/transcript/t
head(df)
```

year	semester	course_number	credits	letter_grade	numerical_value
Freshman	Fall	COMP100	3	A-	3.7
Freshman	Fall	COMP110	3	A	4.0
Freshman	Fall	HUMA181	3	A-	3.7
Freshman	Fall	SOCI181	3	A-	3.7
Freshman	Fall	ELEC181	3	B+	3.3
Freshman	Spring	COMP120	3	A	4.0

5 Measures of Central Tendency - Mean & Median

- **Mean**: The average of a set of numbers, $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$

```
mean(df$numerical_value)
```

```
[1] 3.78
```

- **Weighted Mean**: The mean where some values contribute more than others, $\bar{x}_w = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$

```
sum(df$numerical_value * df$credits)/sum(df$credits)
```

```
[1] 3.790244
```

```
weighted.mean(df$numerical_value, w = df$credits)
```

```
[1] 3.790244
```

- **Trimmed Mean**: The mean after removing a specified number of the highest and lowest values.

```r
mean(df$numerical_value, trim = 0.1)
```

```
[1] 3.8125
```

- **Median**: The middle value in a sorted list of numbers

```r
median(df$numerical_value)
```

```
[1] 3.85
```

6 Measures of Central Tendency - Mode

- **Mode**: The value that appears most frequently in a set.
- As discussed before, mode is more appropriate for qualitative data values.
- So, let's compute mode for the `letter_grade`
- However, in R, there is no built-in function to compute the mode directly.
- Therefore, we need to install the `DescTools` package

```r
if(!require(DescTools))
  install.packages("DescTools",repos = "http://cran.us.r-project.org")
```

```
Loading required package: DescTools
```

- Now we can run the `Mode()` function from `DescTools` package

```r
library(DescTools)
Mode(df$letter_grade)
```

```
[1] "A"
attr(,"freq")
[1] 20
```

7 Descriptive Statistics using `DescTools`

Measure	Function	Description
Mode	`Mode(data)`	Computes the mode. Returns multiple modes if they exist.
Mean	`Mean(data)`	Computes the arithmetic mean.
Weighted Mean	`WtdMean(data)`	Computes the weighted mean.
Median	`Median(data)`	Computes the median.
Trimmed Mean	`Mean(data, trim)`	Computes trimmed mean. `trim` is fraction (0 to 0.5) of observations to be trimmed.
Standard Deviation	`Std(data)`	Computes the sample standard deviation.
Variance	`Var(data)`	Computes the variance.
Range	`Range(data)`	Computes the range (difference between max and min).
Interquartile Range	`IQR(data)`	Computes the interquartile range.

8 Measures of Spread - Range

- **Range**: Difference between the largest and smallest values,

$$\text{Range} = x_{\max} - x_{\min}$$

```r
max(df$numerical_value) - min(df$numerical_value)
```

```
[1] 0.7
```

```r
range(df$numerical_value) # The base R (built-in)
```

```
[1] 3.3 4.0
```

```r
Range(df$numerical_value) # From DescTools
```

```
[1] 0.7
attr(,"bounds")
[1] 3.3 4.0
```

9 Measures of Spread - IQR

- **Interquartile Range (IQR)**: Difference between the first and third quartiles,

$$\text{IQR} = Q_3 - Q_1$$

  - First Quartile:
    - Also known as the lower quartile or the 25th percentile.
    - It is the value below which 25% of the data falls. In other words, it cuts off the lowest 25% of the data.
  - Third Quartile:
    - Also known as the upper quartile or the 75th percentile.
    - It is the value below which 75% of the data falls, meaning it cuts off the lowest 75% of data points.

```r
quantile(df$numerical_value)
```

```
  0%  25%  50%  75% 100%
3.30 3.70 3.85 4.00 4.00
```

```r
quantiles = quantile(df$numerical_value)
quantiles[4] - quantiles[2]
```

```
75%
0.3
```

```r
IQR(df$numerical_value)
```

```
[1] 0.3
```

10 Measures of Spread - Standard Deviation & Variance

- **Standard Deviation**: Measures the amount of variation or dispersion of a set of values,

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

```r
sd(df$numerical_value)
```

```
[1] 0.2613574
```

- **Variance**:

$$var(x) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} = s^2$$

```r
s = sd(df$numerical_value)
s^2
```

```
[1] 0.06830769
```

```r
Var(df$numerical_value) # From DescTools
```

```
[1] 0.06830769
```

11 Measures of Spread - Mean Absolute Deviation (MAD)

- **Mean Absolute Deviation**: (**MAD**) a measure of dispersion representing the average distance of each data point from the mean

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|x_i - \bar{x}|$$

```r
mean(abs(df$numerical_value - mean(df$numerical_value)))
```

```
[1] 0.22
```

- MAD is sensitive to outliers.

12 Exericse - Major GPA *vs.* non-Major GPA

- Using the provided dataset, compare the GPA (the `numerical_value` column) of the student in their major courses versus the non-major courses. For this dataset, Computer Science courses are the major courses, and their course numbers start with `"COMP"`.
- Hint: You might find the `startsWith()` function in R useful to filter rows based on the course number.

13 Solution - Major GPA *vs.* non-Major GPA

- Major Courses

We can search for the rows with major courses using `startsWith()`

```
flag = startsWith(df$course_number, "COMP")
flag
```

```
 [1]  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
[13] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
[25] FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
[37]  TRUE FALSE FALSE FALSE
```

- `TRUE` : a major course
- `FALSE` : a non-major course

```
major_courses = df[flag, ]
head(major_courses)
```

	year	semester	course_number	credits	letter_grade	numerical_value
1	Freshman	Fall	COMP100	3	A-	3.7
2	Freshman	Fall	COMP110	3	A	4.0
6	Freshman	Spring	COMP120	3	A	4.0
7	Freshman	Spring	COMP130	3	A-	3.7
11	Sophomore	Fall	COMP200	4	A	4.0
12	Sophomore	Fall	COMP210	4	A	4.0

```
major_gpa = median(major_courses$numerical_value)
major_gpa
```

```
[1] 4
```

- Non-Major Courses

To filter for non-major courses, we can just *negate* `flag` i.e.,

```
!flag
```

```
 [1] FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
[13]  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
[25]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE
[37] FALSE  TRUE  TRUE  TRUE
```

- `TRUE` : a non-major course
- `FALSE` : a major course

```
nonmajor_courses = df[!flag, ]
head(nonmajor_courses)
```

	year	semester	course_number	credits	letter_grade	numerical_value
3	Freshman	Fall	HUMA181	3	A-	3.7
4	Freshman	Fall	SOCI181	3	A-	3.7
5	Freshman	Fall	ELEC181	3	B+	3.3
8	Freshman	Spring	HUMA191	4	B+	3.3
9	Freshman	Spring	SOCI191	2	A-	3.7
10	Freshman	Spring	ELEC191	3	A-	3.7

```
nonmajor_gpa = median(nonmajor_courses$numerical_value)
nonmajor_gpa
```

```
[1] 3.7
```

- Using the `summary()` function

```
summary(major_courses$numerical_value)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.3	3.7	4	3.8375	4	4

```
summary(nonmajor_courses$numerical_value)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.3	3.7	3.7	3.741667	4	4