Today, we are going to cover summarize data numerically.

Instead of looking at each number individually, we would like to summarize the data using a few numbers to help us understand the big picture of the given data set.

Let’s load the dataset and look at the variable names:

```
library(readxl)
StudentsPerformance <- read_excel("C:/Users/yliu3/OneDrive - Maryville Univer
sity/Online DSCI502 R Programming/DataSets/StudentsPerformance.xlsx")

colnames(StudentsPerformance)

## [1] "Gender"                   "Race"
## [3] "ParentalLevelOfEducation" "Lunch"
## [5] "TestPreparationCourse"    "MathScore"
## [7] "ReadingScore"             "WritingScore"
```

When we look at these variable names/column names, we find that there are two cases:

- Continuous/numerical variables have infinite many cases in theory. In theory, we cannot list all of them. For example, math Score could be 90, 90.1, 90.15 etc. Similarly, Reading scores and writing scores are continuous/numerical variables.
- Factors (categorical variables) have only finite many cases; we can list all of them. For example Gender has only two cases; female and male in the given data set. Similarly, Race (5 cases), Parental level of Education (6 cases), Lunch (2 cases) and Test Preparation Course (2 cases) are factors.

To test whether a column is a continuous variable, we can use one of the following R functions:

- is.numeric() returns TRUE if the input is a continuous variable.
- class() returns numerical if the input is a continuous variable.

For example, to test whether the MathScore variable is continuous or not, we can use the following R command:

```
is.numeric(StudentsPerformance$MathScore)

## [1] TRUE
```

Since is.numeric returns TRUE, this means that math score variable is continuous.

We can also use class function too.

```
class(StudentsPerformance$MathScore)

## [1] "numeric"
```

Similarly, class function returns "numeric", which means that the math score variable is continuous.

Let's find its' maximum and minimum using the following codes respectively:

```
max(StudentsPerformance$MathScore)

## [1] 100

min(StudentsPerformance$MathScore)

## [1] 0
```

We can also find its' maximum and minimum using range () function.

```
range(StudentsPerformance$MathScore)

## [1]   0 100
```

Therefore, the maximum and minimum of the math scores are 100 and 0 respectively.

Sometimes we are interested in the range, which is the difference between the maximum and minimum. To find it, we can use two methods in R.

First method is based on range function.

```
math.range <- range(StudentsPerformance$MathScore)
math.range[2]-math.range[1]
```

range() function returns the minimum and maximum of a given data set, then we subtract minimum from maximum.

```
Second method is based on max and min functions.

max(StudentsPerformance$MathScore) -min(StudentsPerformance$MathScore)

## [1] 100
```

We can see that it has a very wide range of 100.

To aggregate the data, we typically consider the average/mean of a numerical/continuous variable. To compute the average of a **continuous** variable, we can use the **mean()** function in R to find the average of the numerical values

Let’ run the R code:

```r
mean(StudentsPerformance$MathScore)

## [1] 66.089
```

The mean() is a good function to summarize the numerical data with similar quantities into a single number. But when the data has extreme values, the mean may be misleading.

To reduce the impact of the outlier, we can use the median to summarize the data since the outliers are typically ignored in the median

The median() function in R returns the median of a numeric vector. To find the median of the math score, we can use the following R codes:

```r
median(StudentsPerformance$MathScore)

## [1] 66
```

You may notice that the median and mean of the math scores are very close due to the math scores having a symmetric distribution. But in general the mean and median could be very different due to some outliers.

The mean and average of numerical values only summarize the “center” of the data set. To have a complete picture of data, we also need to consider the dispersion of numerical values relative to its mean. This dispersion is called the standard deviation in statistics.

To find the standard deviation in R, we need to use sd() function. For example, to find the standard deviation of math scores, we may use the following code:

```r
sd(StudentsPerformance$MathScore)

## [1] 15.16308
```