Today, I am going to cover percentile and correlation.

To have a big picture of numerical values, we often use percentiles to report them.

There are several numerical methods to compute the given percentile. These results are very similar when we have a "larger" data set. They are very technical; we will not cover it in this courses. The following special percentiles are often used:

- 0th percentile, called the minimum, often denoted by $\min$,which means that there are 0% of the data less than $\min$
- 25th percentile, called first quartile, often denoted by $Q1$, which means that there are 25% of the data less than or equal to $Q1$
- 50th percentile, called second quartile or **median**, often denoted by $Q2$, which means that there are 50% of the data less than or equal to $Q2$. It also means that there are 50% of the data larger than or equal to $Q2$. Therefore $Q2$ is the median of the data set.
- 75th percentile, called third quartile, often denoted by $Q3$, which means that there are 75% of the data less than or equal to $Q3$
- 100th percentile, called the maximum, often denoted by $\max$, which means there are 100% of the data less than or equal to $\max$.

To find the percentile in R, we need to use the quantile() function and specify the corresponding percentiles. For example, we can use the following R code to find the five percentiles above:

```
quantile(StudentsPerformance$MathScore, probs = c(0, 0.25, 0.5, 0.75, 1))
```

```
##   0%  25%  50%  75% 100%
##    0   57   66   77  100
```

Note that the first argument is the data column, the second argument is the percentiles (probabilities) between 0 and 1.

Of course, we can find other percentiles by providing corresponding probabilities. For example, we can find the 10th, 20th, 30th, …, 90th percentile using the following R code:

```
quantile(StudentsPerformance$MathScore, probs = seq(from = 0.1, to = 0.9, by
= 0.1))
```

```
## 10% 20% 30% 40% 50% 60% 70% 80% 90%
##  47  53  59  62  66  70  74  79  86
```

Note that we use the seq() function to specify the corresponding probabilities instead of listing all of them. This sequence function has three parameters:

- Starting point denoted by "from"" parameter
- Ending point denoted by "to"" parameter

- Step size denoted by “by” parameter

Correlation

So far, we have only considered one numerical variable. When we build machine learning models, we are very interested in the relationship between two numerical values. The correlation coefficient summarizes the trend between the two numerical values using mean and standard deviation.

It can be shown that the correlation coefficient is always between -1 and 1 due to the normalization by the two standard deviations in the denominator. There are three important cases to consider:

- Positive correlation coefficient means when one continuous variable increases (deceases), the second numerical values increases (deceases).
- Negative correlation coefficient means when one continuous variable increases (deceases), the second numerical values decease (increases).
- Zero correlation coefficient means there is no relationship between these two numerical variables at all.

To compute the correlation coefficient in R, we use the cor() function. For example, we can compute the correlation coefficient between the math score and reading score using the following R code:

```
cor(StudentsPerformance$MathScore,StudentsPerformance$ReadingScore)
```

```
## [1] 0.8175797
```

Since the correlation of coefficient is 0.82. It is positive and very close to 1. This means that there is a very strong relationship between them. If a student’s math score is higher, his/her reading score tends to higher too.