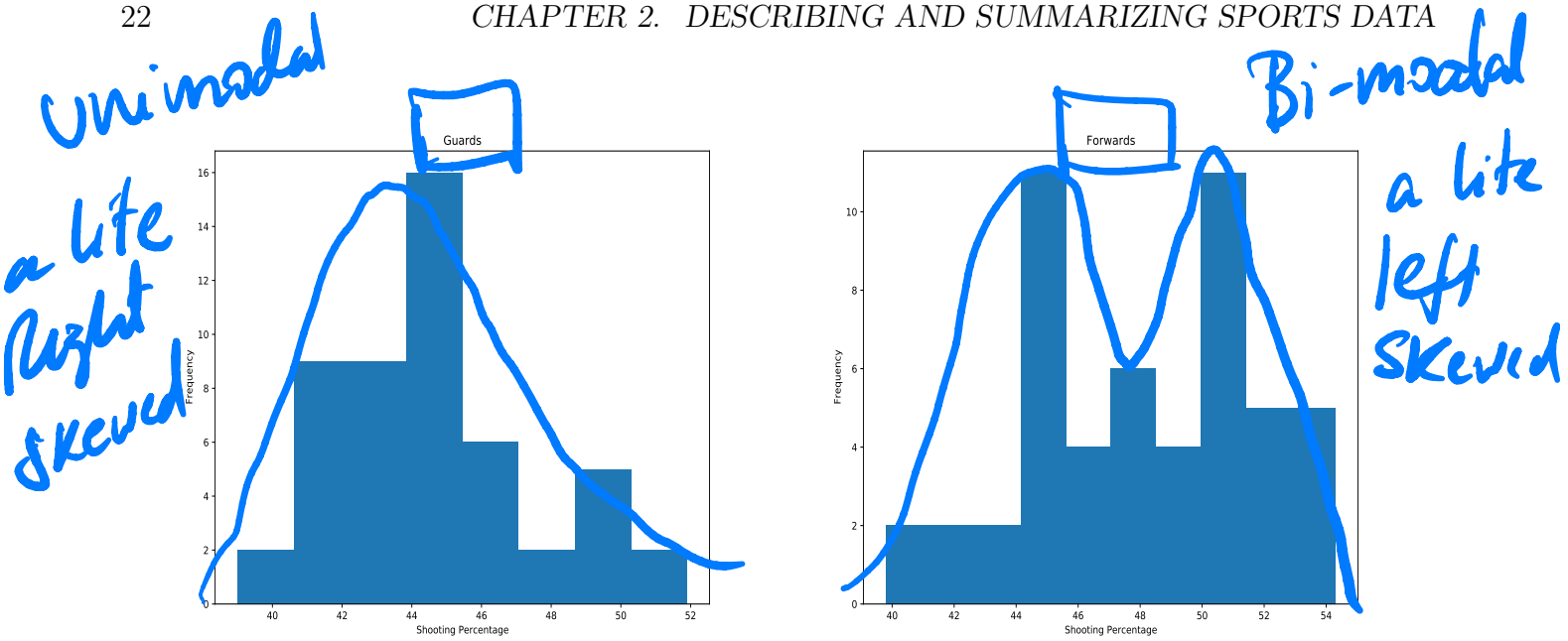


Figure 2.4: Left panel: shooting percentage of guards. Right panel: shooting percentage of forwards.

2.5 Mean and Median

Although a frequency distribution provides some summarization of a dataset, in some cases a single number summary is useful. For quantitative data, the *mean* and the *median* are the most commonly used summaries of this type. The mean of a numeric dataset is simply the average, that is, the total sum divide by the number of observations. For example, consider the data on Tom Brady's passing yards in his 159 starts (2001-2011), the mean is 251.1 yards per game. Averages are commonly used in traditional sports statistics. The following box shows the R code that was used compute the average yards per game of Tom Brady's dataset.

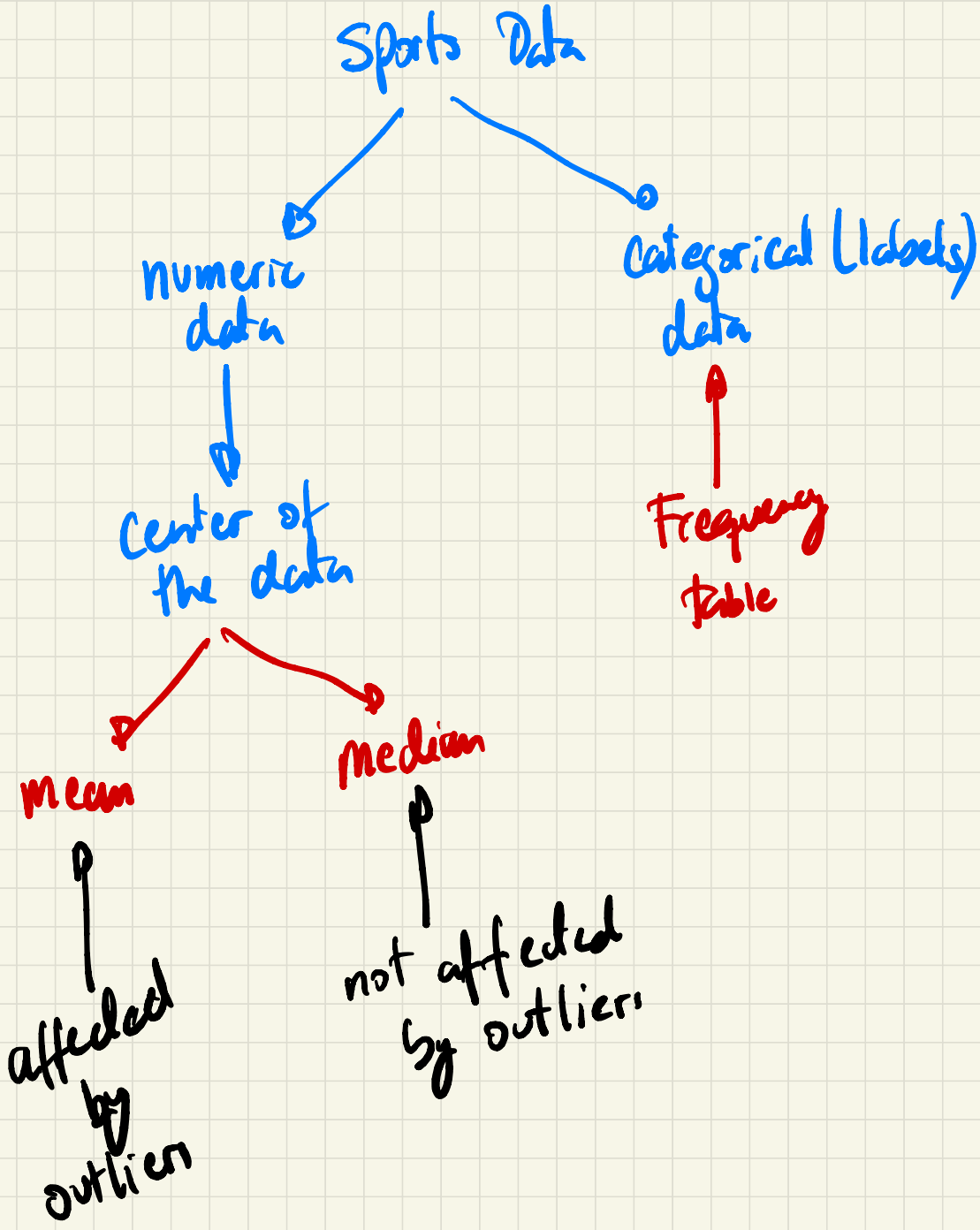
R code

```
## Reading Tom Brady's passing yards data (2001-2011)
brady = read.csv(file = 'Dataset_2_1.csv')

## Computing the mean of passing yards
mean(brady$PY)
```

The following box shows the Python code that was used compute the average yards per game of Tom Brady's dataset.

Python code

```
import pandas as pd
```


Sports Data
numeric data
categorical (labels) data
center of the data
Frequency Table
mean
median
affected by outliers
not affected by outliers

Python code

```
## Reading Tom Brady's passing yards data (2001-2011)
brady = pd.read_csv('Dataset_2_1.csv')

## Computing the mean of passing yards
brady['PY'].mean()
```

The median is often a useful alternative to the mean. The median is found by putting the data values in order and then finding the middle value. Thus, the median has the interpretation that half of the data values are above the median and half of the data values are less than the median. For the Brady's passing yard dataset, the median value is 249. The following box shows the R code that was used to compute the median of the Tom Brady's passing yard dataset.

R code

```
## Reading Tom Brady's passing yards data (2001-2011)
brady = read.csv(file = 'Dataset_2_1.csv')

## Computing the mean of passing yards
median(brady$PY)
```

The following box shows the Python code that was used compute the median yards per game of Tom Brady's dataset.

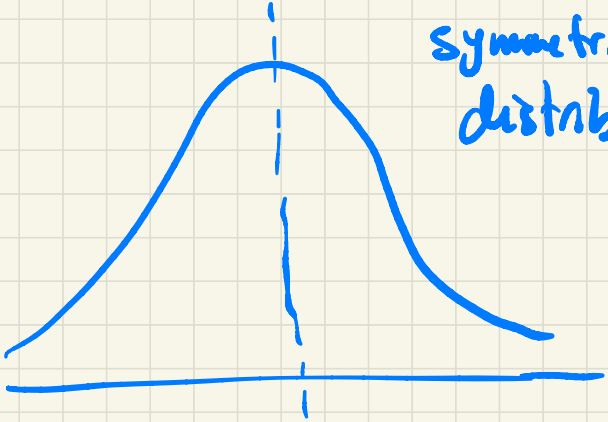
Python code

```
import pandas as pd

## Reading Tom Brady's passing yards data (2001-2011)
brady = pd.read_csv('Dataset_2_1.csv')

## Computing the mean of passing yards
brady['PY'].median()
```

The mean and median are two different ways to summarize a dataset, and which is better in a given situation depends on the goals of the analysis. If the distribution is approximately symmetric, then the mean and the median are approximately equal. For skewed distribution, however, the mean and the median are different because very large (or very small) observations have a greater effect on the mean than on the median.

2.6 Measuring Variation

Variation is an important part of all sports. The players in a league have varying skill sets, and their game performance vary. If two teams play each other several times, the outcomes will vary.



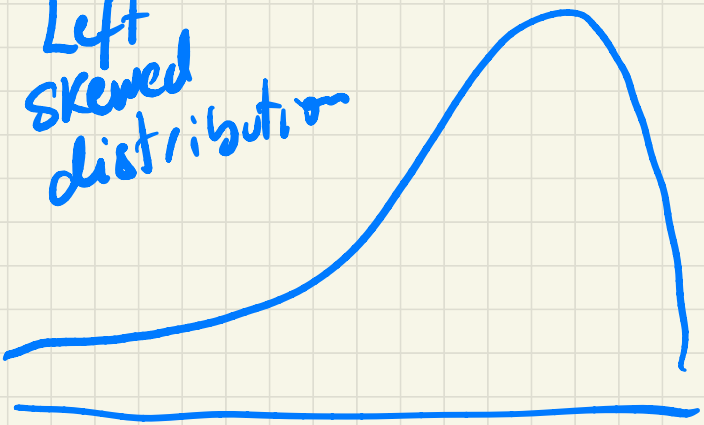
symmetric distribution

$\Rightarrow$ mean $\approx$ median

approx

Right skewed distribution

$\Rightarrow$ mean > median



Left skewed distribution

$\Rightarrow$ mean < median

Understanding, and accounting for, variation is a central goal of analytical methods. In this section, we will discuss standard measures that are commonly used to quantify variation. In later chapters, we will discuss other measures that are used to quantify variation in sports.

The variation of numerical variables refers to how the observed values of the variable differ from one another. One approach to measuring variation is to choose a reference value for the data and consider how the values differ from the the reference value. A commonly used choice for the reference value is the mean of the data.

The *standard deviation* of a dataset is, roughly speaking, the average distance from an observation to the mean of the data values. To compute the standard deviation, we first average the squared distances of the data values from the mean value; this is called the *variance* of the dataset. Although variances are sometimes used directly, they have the drawback that the units are the square of the units the observations themselves. For instance, if the measurements is in yards, the units of the variance will be yards squared. Therefore, we typically use the square root of the variance, which is called the standard deviation. The standard deviation may be viewed as the "typical" distance of a data value to the mean of the dataset. Note that the units of the standard deviation are the same as the units of the variable under consideration.

For example, consider the shooting percentage dataset of 2010-2011 NBA by position. The standard deviation of forwards is 3.66 while the standard deviation of guards is 2.88. The following box shows the R code that was used to compute the variance and standard of the shooting percentage.

R code

```
## Reading csv file
shooting = read.csv(file = 'Dataset_2_3.csv')

## Computing the variance of shooting percentage of forwards
var(shooting$SPCT[shooting$Pos %in% c('SF', 'PF')])

## Computing the variance of shooting percentage of guards
var(shooting$SPCT[shooting$Pos == 'G'])

## Computing the standard deviation of shooting percentage of forwards
sd(shooting$SPCT[shooting$Pos %in% c('SF', 'PF')])

## Computing the standard deviation of shooting percentage of guards
sd(shooting$SPCT[shooting$Pos == 'G'])
```

The following box shows the Python code that was used to compute the variance and standard of the shooting percentage.

Python code

```
import pandas as pd
```

Python code

```
import numpy as np

## Reading the csv file
shooting = pd.read_csv('Dataset_2_3.csv')

## Computing the variance of shooting percentage of forwards
var_forward = np.var(shooting['SPCT'][np.isin(shooting['Pos'], ['PF', 'SF'])])

## Computing the variance of shooting percentage of guards
var_guard = np.var(shooting['SPCT'][shooting['Pos'] == 'G'])

## Computing the standard deviation of shooting percentage of forwards
sd_forward = np.std(shooting['SPCT'][np.isin(shooting['Pos'], ['PF', 'SF'])], ddof = 1)

## Computing the standard deviation of shooting percentage of guards
sd_guard = np.std(shooting['SPCT'][shooting['Pos'] == 'G'], ddof = 1)
```

One drawback of the standard deviation is that it is difficult to interpret directly. One reason for the difficulty in interpreting the standard deviation is that it is sensitive to the measurement units used; specially, if the data values are all multiplied by a constant k, then the standard deviation of the new dataset is k times the original standard deviation.

To eliminate the dependence on the units used, it is sometimes useful to express the standard deviation as a proportion, or percentage, of the average data values. The *coefficient of variation* (CV) is the standard deviation divided by the mean. Note that the coefficient of variation is a dimensionless quantity; therefore, it does not depend on the measurement scale. If all data values are multiplied by a constant k, both the standard deviation and the average of the new data values are multiplied by k and k cancels when taking the ratio. The coefficient of variation is generally only used for variables in which the values can't be negative or zero. Table 2.7 shows the the coefficients of variation for shooting percentage of guards and forwards in 2010-2011 NBA season.

Table 2.7: Coefficient of Variation for shooting percentage in 2010-2011 NBA season

Position	CV(%)
Guard	6.5%
Forward	7.6%

Based on these results, forwards have the most variability in shooting percentage. The following box shows the R code that was used to compute the coefficient of variation for guards and forwards in 2010-2011 NBA season.

R code

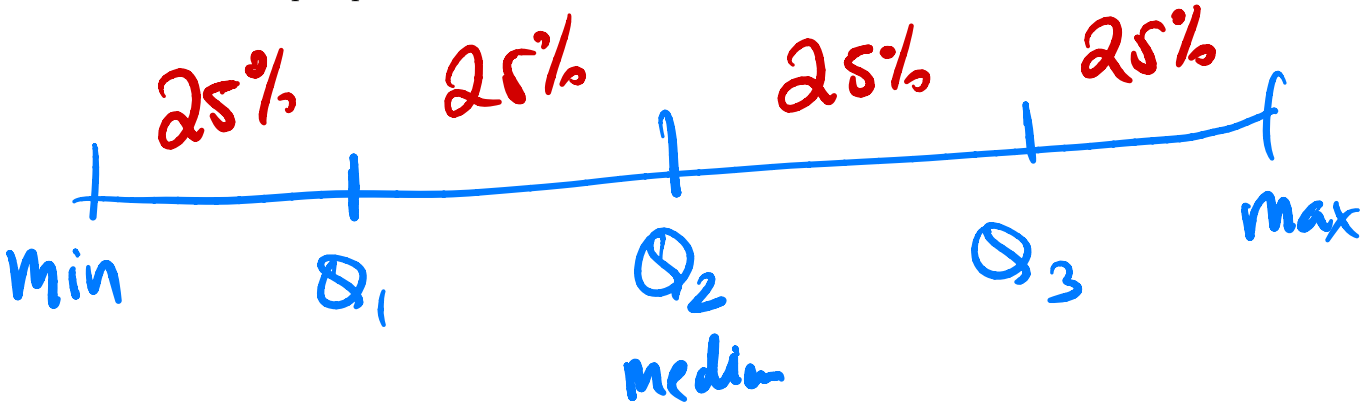
```
## Computing the CV of shooting percentages of forwards
mean_forward = mean(shooting$SPCT[shooting$Pos %in% c('SF', 'PF')])
sd_forward = sd(shooting$SPCT[shooting$Pos %in% c('SF', 'PF')])
CV_forward = sd_forward / mean_forward

## Computing the CV of shooting percentages of guards
mean_guard = mean(shooting$SPCT[shooting$Pos == 'G'])
sd_guard = sd(shooting$SPCT[shooting$Pos == 'G'])
CV_guard = sd_guard / mean_guard
```

The following box shows the Python code that was used to compute the coefficient of variation for guards and forwards in 2010-2011 NBA season.

Python code

```
## Computing the CV of shooting percentages of forwards
mean_forward = np.mean(shooting['SPCT'][np.isin(shooting['Pos'], ['PF', 'SF'])])
sd_forward = np.std(shooting['SPCT'][np.isin(shooting['Pos'], ['PF', 'SF'])], ddof = 1)
CV_forward = sd_forward / mean_forward

## Computing the CV of shooting percentages of guards
mean_guard = np.mean(shooting['SPCT'][shooting['Pos'] == 'G'])
sd_guard = np.std(shooting['SPCT'][shooting['Pos'] == 'G'], ddof = 1)
CV_guard = sd_guard / mean_guard
```

Another approach to measuring the variation of a variable is to look at the spread of the values of the dataset. One such measure is the *range*, which is defined as the difference between the maximum value minus the minimum value in the dataset. However, the range is too sensitive to extreme values. An alternative approach is to base a measure of variation on the *quartiles* of the data. In the same way that the median is the *midpoint* of the dataset, the three quartiles divide the data into four equal parts.



Interquartile Range (IQR) $= Q_3 - Q_1$