```
[1] 1952
```

```
> min( year, data=swim )
```

```
[1] 1905
```

The data= argument tells the function which data frame to pull the variable from. Leaving off that argument leads to an error:

```
> mean( year )
```

```
Error in mean(year) :
   error in evaluating the argument 'x', object 'year' not found
```

Of course, you know that the variable year is defined within the data frame swim, but you have to give this information to the function mean() explicitly, otherwise it doesn't know where to find the variable.

Take note that many long-time users of R use the $ notation, like this:

```
> mean( swim$year )
```

Think of this as referring to the variable by both its family name (the data frame's name,"swim") and its given name ("year"), something like Einstein$Albert.

The advantage of the data= approach becomes evident when you construct statements that involve more than one variable within a data frame. For instance, here's a calculation of the mean year separately for the different sexes:

```
> mean( year ~ sex, data=swim )
```

```
   F    M
1951 1954
```

You will see much more of the ~ ("tilde") starting in Chapter 4. It's the R notation for "modeled by," or "broken down by," or "versus."

Both the mean() and min() functions have been arranged by the mosaic package to look in the data frame when interpreting variables, but not all R functions — particularly older ones — are designed this way. For instance:

```
> sqrt(year, data=swim )
```

```
Error in try(sqrt(year, data = swim)) : object 'year' not found
```

When you encounter a function that can't handle the data= format, you can use the special with() function that instructs any function to refer to a data frame:

```
> with( data=swim, sqrt(year) )
```

```
 [1] 43.6 43.7 43.7 43.7 43.8 43.8 43.8 43.9 44.0 44.0 44.0 44.1 44.1 44.1 44.2
[16] 44.2 44.3 44.3 44.4 44.4 44.4 44.4 44.4 44.5 44.5 44.6 44.6 44.6 44.7 44.7
[31] 44.7 43.7 43.7 43.7 43.7 43.8 43.8 43.9 43.9 43.9 43.9 43.9 43.9 44.0 44.0
[46] 44.0 44.2 44.2 44.3 44.3 44.3 44.4 44.4 44.4 44.5 44.5 44.5 44.6 44.6 44.7
[61] 44.7 44.8
```