Chapter 2: Summarizing data

Introductory Statistics for Engineering Experimentation

Peter R. Nelson, Marie Coffin and Karen A.F. Copeland

Slides by Douglas Bates

Outline

Section 2.1: Univariate data

- Graphs can
  - summarize the data
  - show typical and atypical values
  - highlight relationships between variables
  - show how the data are spread out, which we call the *distribution* of the data
- Often we observe multiple characteristics on each experimental run or observation. We call such data *multivariate*. If we only observe one characteristic we say the data are *univariate*.
- Typical plots of univariate, numeric data are *histograms* (`histogram`) or *density plots* (`densityplot`), *box-and-whisker plots* (`bwplot`) and *dotplots* (`dotplot`). (Names in parentheses are the names of the corresponding *R* function from the `lattice` package.)

The railcar data

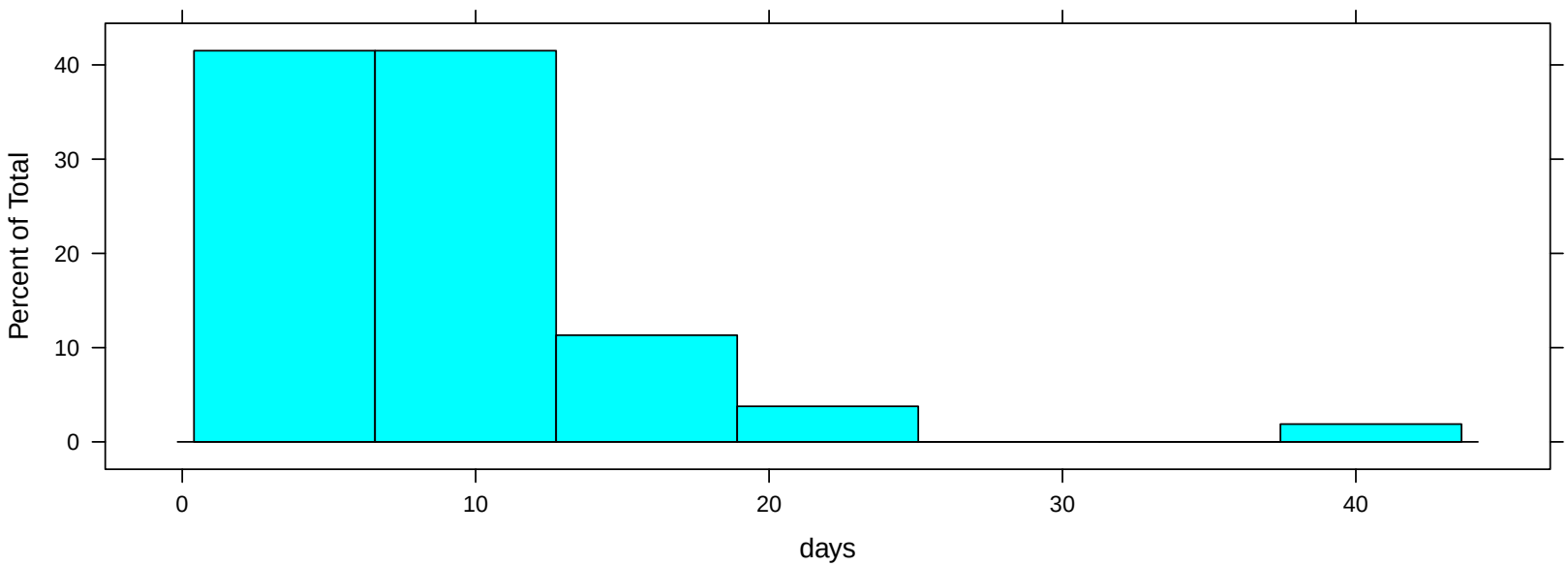
- Example 2.1.1 describes the `railcar` data, which are 53 observations of the number of days that a customer holds a rail car after delivery. The authors say the data are in a file named *railcar.txt*. We will use the dataset called `railcar` from the `EngrExpt` package for *R*.
- All of the data sets are described in Appendix A, starting on page 424. When you start *R* you should attach the EngrExpt package, using `library(EngrExpt)`. To check on the form of a data set use, e.g., `str(railcar)`, and compare the result to the description in Appendix A.

```
> library(EngrExpt)
> str(railcar)

'data.frame': 53 obs. of  1 variable:
 $ days: int  4 42 4 4 3 5 5 5 3 7 ...
```

Histograms

- A histogram is a simple bar chart of the number of observations in each of a set of adjacent, constant-width intervals.
- It was popular in the days when all graphics, and most calculations, needed to be done by hand. It shows the distribution of the data (see the comments in example 2.1.1).



Creating a histogram

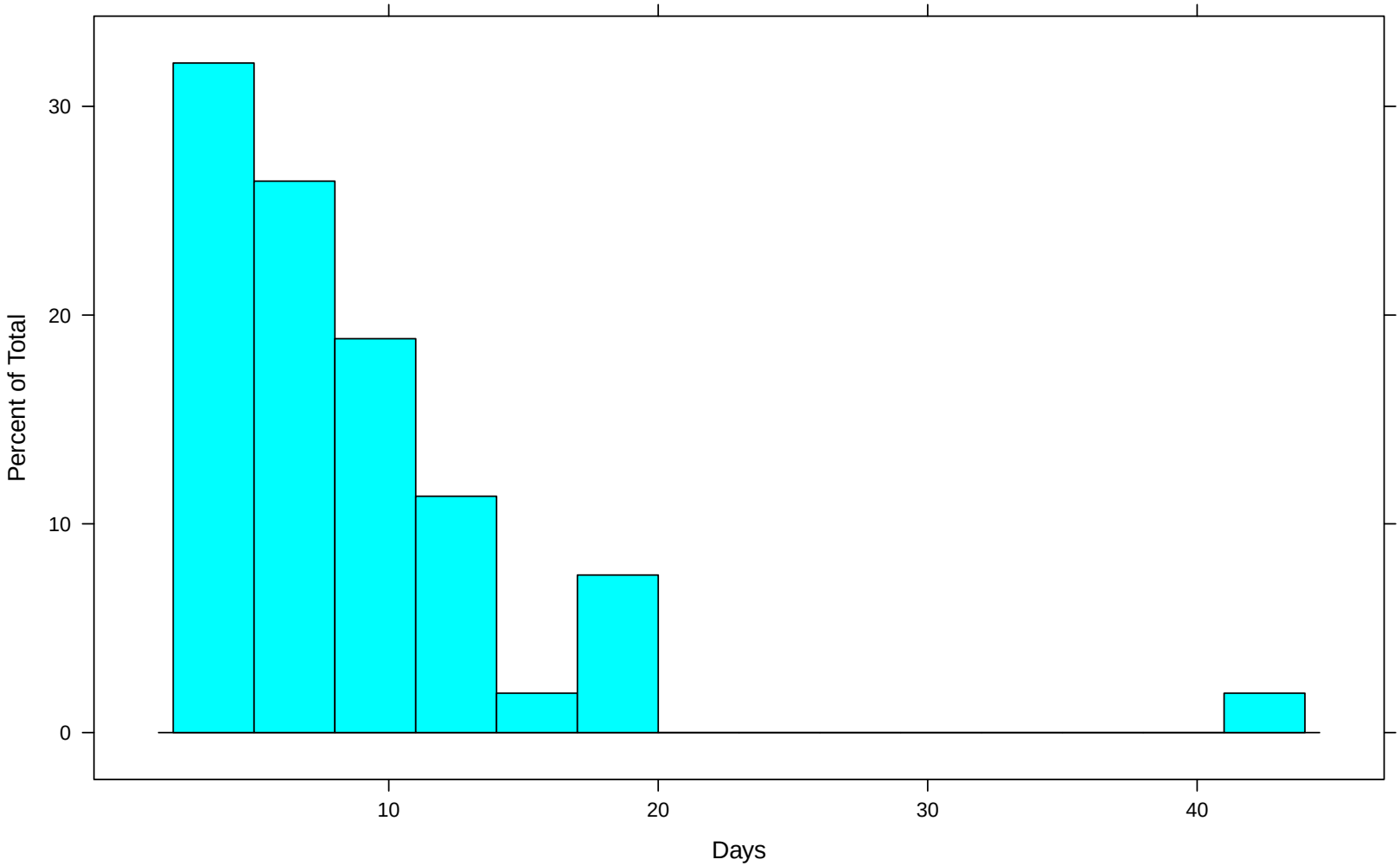
- The text gives a description of creating a histogram by hand.
- In *R*, you can use the `histogram` function. The plot on the previous slide was created with

  ```
  > histogram(~days, railcar)
  ```

- There are several optional arguments for the `histogram` function. To create a plot like Figure 2.1 we specify the break points for the intervals (argument `breaks`) and also change the label on the x-axis (argument `xlab`).

```
> histogram(~days, railcar, breaks = seq(2, 44,
+     3), xlab = "Days")
```

Histogram like Figure 2.1

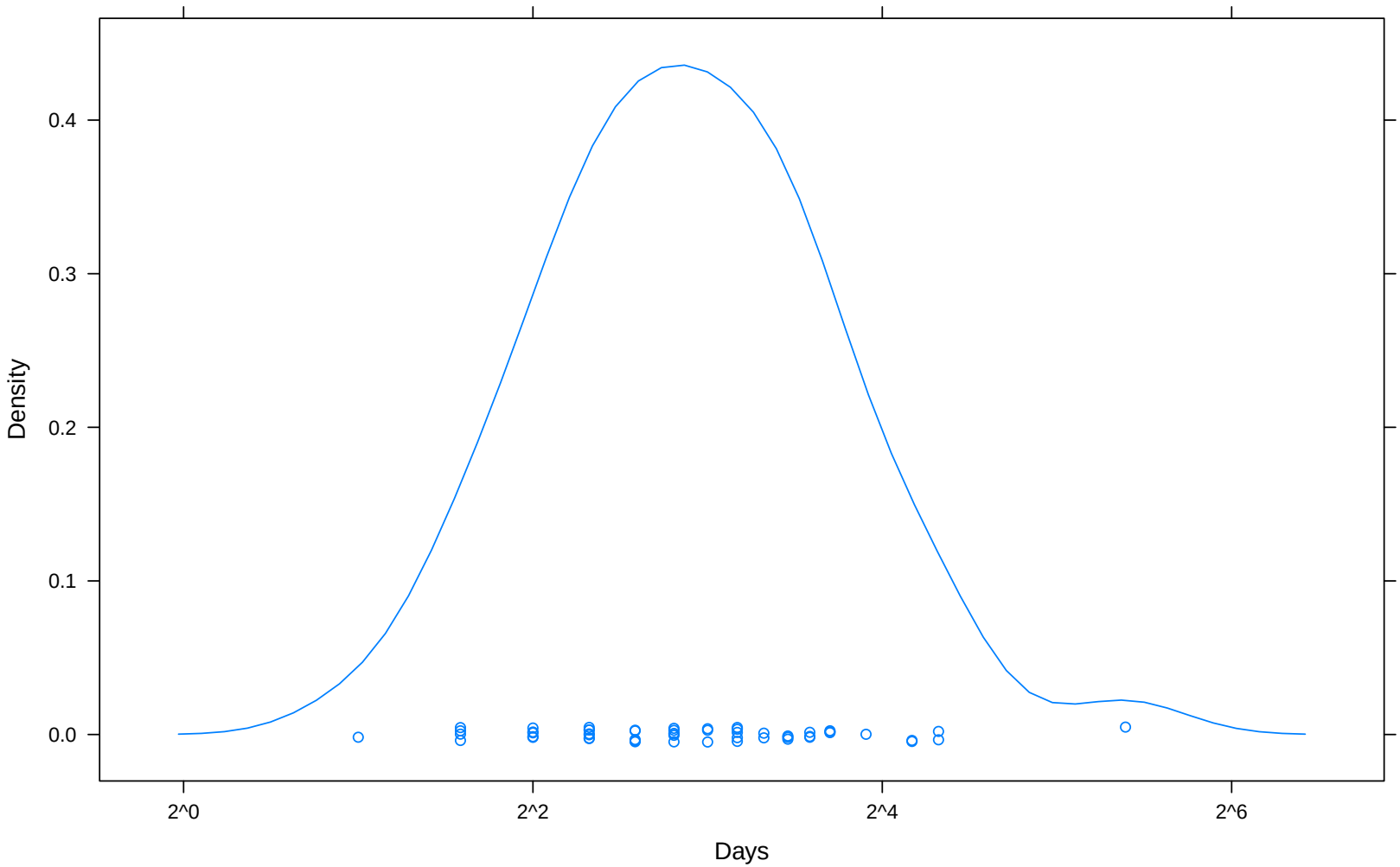


Density plots

- This topic is not covered in the text. It requires more sophisticated software than they were using.
- Notice that the histogram shape depends on the somewhat arbitrary choice of intervals.
- If we are interested in the shape of the distribution of the observations we can use an alternative called a density plot.
- Without going in to details, an empirical density plot centers a narrow, "bell-curve" density at each observed data point and sums the result. The version in *R* also adds a "rug" with the original data values plotted as points and jittered vertically (so you can see multiple data points with the same value).

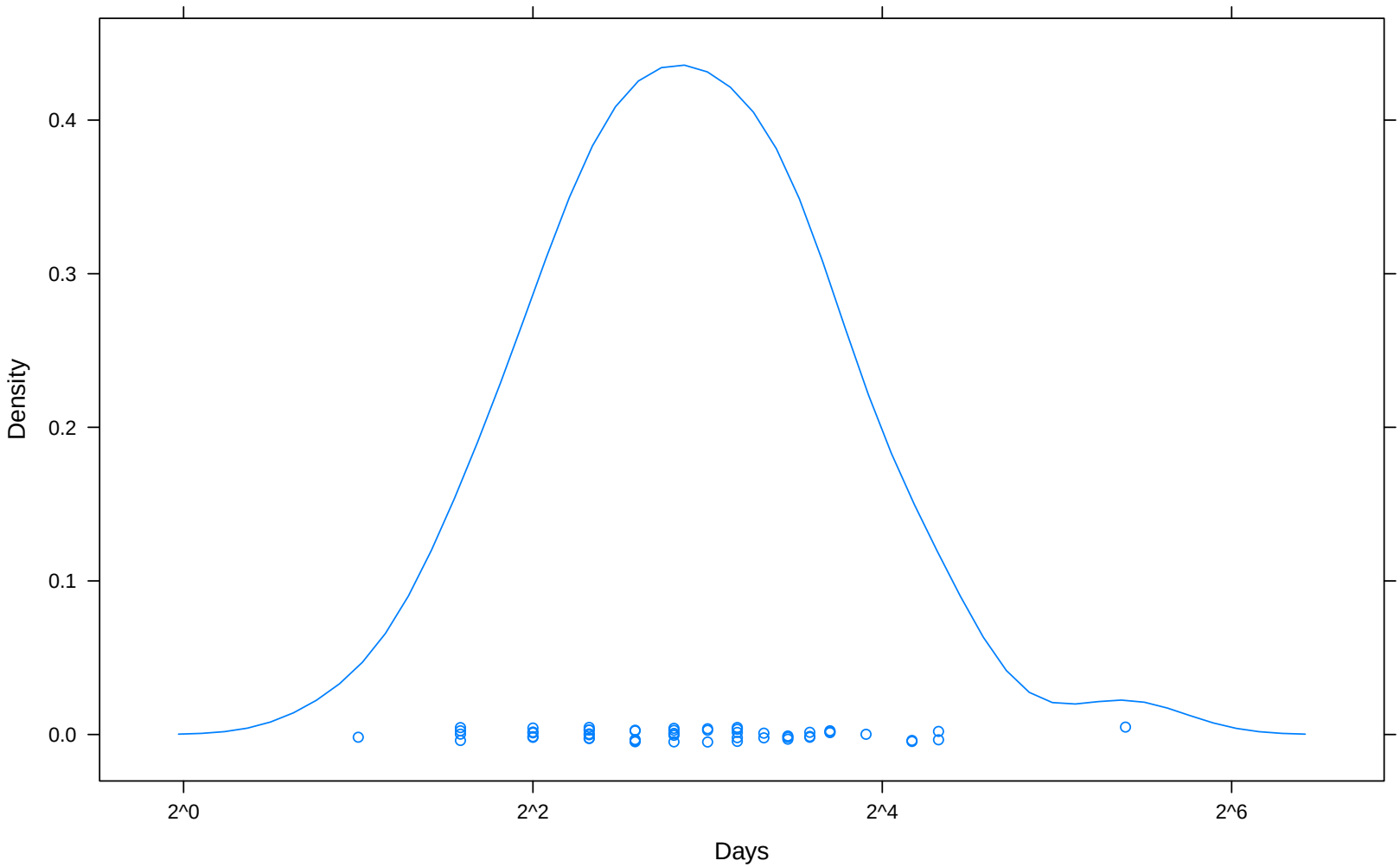
```
> densityplot(~days, railcar, xlab = "Days")
```

Density plot of the rail data.



The skewness of this plot indicates that we may want to consider the logarithm of the days.

Density plot of the rail data (logarithmic scale).



The skewness of this plot indicates that we may want to consider the logarithm of the days.