data frame mile2 and the second argument ase() defines the aesthetics that map variables in the data frame to aspects of the graph. We indicate in the aes argument that Year is to be plotted along the horizontal (x) axis, seconds is to be plotted along the vertical (y) axis, and the variable Gender will be used as the color and shape aesthetics. We store this graph initialization information in the variable p.

```
> library(ggplot2)
> p = ggplot(mile2, aes(x = Year, y = seconds,
+  color = Gender, shape = Gender))
```

The function ggplot does not perform any plotting – it just defines the data frame and the aesthetics. We construct a plot by adding geom and statistic layers to this initial definition. A scatterplot is constructed using the geom_point function; the size = 4 arguments indicates that the points will be drawn twice the usual size of 2. Then we add a smoothing lowess curve using the geom_smooth function.

```
> p + geom_point(size = 4) + geom_smooth()
```

Figure 4.20 displays the resulting graphical display. Since the variable Gender has been assigned to the color and shape aesthetics, note that the male and female record times are plotted using different colors and shapes and a legend is automatically drawn outside of the plotting region. Note that since Gender has a color aesthetic, the smoothing curves are graphed for each gender. The record times appear to be linearly decreasing as a function of year for both genders this time period, although the rate of decrease is much greater for the women times.

Although the "grammar" of the ggplot2 system may seem odd at first look, one can construct attractive and useful graphics using a limited amount of R code. The book Hadley [52] provides a comprehensive description of the ggplot2 package and many graphics examples are displayed on the accompanying website had.co.nz/ggplot2.

Exercises

4.1 (Speed and stopping distance). The data frame cars in the datasets package gives the speed (in mph) and stopping distance (in ft) for 50 cars.

a. Use the plot function to construct a scatterplot of speed (horizontal) against dist (vertical).
b. Revise the basic plot by labeling the horizontal axis with "Speed (mpg)" and the vertical axis with "Stopping Distance (ft)," Add a meaningful title to the plot.
c. Revise the plot by changing the plot symbol from the default open circles to red filled triangles (col="red", pch=17).

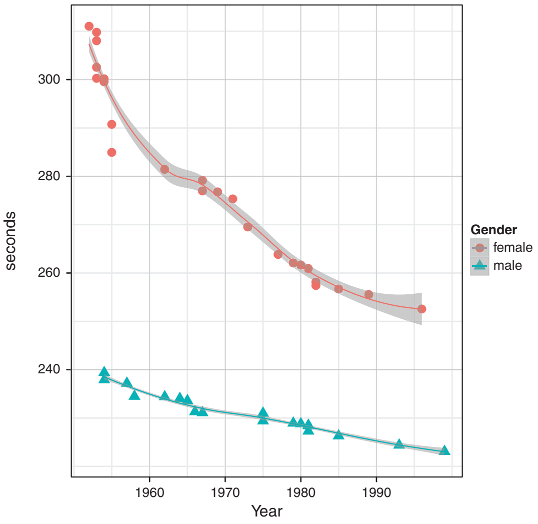

Fig. 4.20 Scatterplots of world record running times for the mile for the men and women using the `ggplot2` package. Smoothing lines are added to show the general pattern of decrease for each gender.

4.2 (Speed and stopping distance (continued)). Suppose that one wishes to compare linear and quadratic fits to the (`speed`,`dist`) observations. One can construct these two fits in R using the code

```
fit.linear = lm(dist ˜ speed, data=cars)
fit.quadratic = lm(dist ˜ speed + I(speedˆ2), data=cars)
```

a. Construct a scatterplot of speed and stopping distance.
b. Using the `abline` function with argument `fit.linear`, overlay the best line fit using line type “dotted” and using a line width of 2.
c. Using the `lines` function, overlay the quadratic fit using line type “long-dash” and a line width of 2.
d. Use a legend to show the line types of the linear and quadratic fits.
e. Redo parts (a) - (d) using two contrasting colors (say red and blue) for the two different fits.

4.3 (Speed and stopping distance (continued)).

a. Construct a residual plot for the linear fit by typing

```
plot(cars$speed, fit.linear$residual)
```

b. Add a blue, thick (`lwd=3`) horizontal line to the residual plot using the `abline` function.
c. There are two large positive residuals in this graph. By two applications of the `text` function, label each residual using the label “POS” in blue.

d. Label the one large negative residual in the graph with the label "NEG" in red.
e. Use the identify function to find the row numbers of two observations that have residuals close to zero.

4.4 (Multiple graphs). The dataset mtcars contains measurements of fuel consumption (variable mpg) and other design and performance characteristics for a group of 32 automobiles. Using the mfrow argument to par, construct scatterplots of each of the four variables disp (displacement), wt (weight), hp (horsepower), drat (rear axle ratio) with mileage (mpg) on a two by two array. Comparing the four graphs, which variable among displacement, weight, horsepower, and rear axle ratio has the strongest relationship with mileage?

4.5 (Drawing houses). The following function house plots an outline of a house centered about the point (x, y):

```
house=function(x, y, ...){
  lines(c(x - 1, x + 1, x + 1, x - 1, x - 1),
     c(y - 1, y - 1, y + 1, y + 1, y - 1), ...)
  lines(c(x - 1, x, x + 1), c(y + 1, y + 2, y + 1), ...)
  lines(c(x - 0.3, x + 0.3, x + 0.3, x - 0.3, x - 0.3),
     c(y - 1, y - 1, y + 0.4, y + 0.4, y - 1), ...)
}
```

a. Read the function house into R.
b. Use the plot.new function to open a new plot window. Using the plot.window function, set up a coordinate system where the horizontal and vertical scales both range from 0 to 10.
c. Using three applications of the function house, draw three houses on the current plot window centered at the locations (1, 1), (4, 2), and (7, 6).
d. Using the ... argument, one is able to pass along parameters that modify attributes of the line function. For example, if one was interested in drawing a red house using thick lines at the location (2, 7), one can type

```
house(2, 7, col="red", lwd=3)
```

Using the col and lty arguments, draw three additional houses on the current plot window at different locations, colors, and line types.
e. Draw a boundary box about the current plot window using the box function.

4.6 (Drawing beta density curves). Suppose one is interesting in displaying three members of the beta family of curves, where the beta density with shape parameters a and b (denoted by Beta(a,b)) is given by

$$f(y) = \frac{1}{B(a,b)} y^{a-1}(1-y)^{b-1},\ 0 < y < 1.$$

One can draw a single beta density, say with shape parameters $a = 5$ and $b = 2$, using the curve function:

```
curve(dbeta(x, 5, 2), from=0, to=1))
```

a. Use three applications of the `curve` function to display the Beta(2, 6), Beta(4, 4), and Beta(6, 2) densities on the same plot. (The `curve` function with the `add=TRUE` argument will add the curve to the current plot.)
b. Use the following R command to title the plot with the equation of the beta density.

   ```
   title(expression(f(y)==frac(1,B(a,b))*y^{a-1}*(1-y)^{b-1}))
   ```

c. Using the `text` function, label each of the beta curves with the corresponding values of the shape parameters *a* and *b*.
d. Redraw the graph using different colors or line types for the three beta density curves.
e. Instead of using the `text` function, add a legend to the graph that shows the color or line type for each of the beta density curves.

4.7 (`lattice` graphics). The dataset `faithful` contains the duration of the eruption (in minutes) `eruptions` and the waiting time until the next eruption `waiting` (in minutes) for the Old Faithful geyser. One is interested in exploring the relationship between the two variables.

a. Create a factor variable `length` that is “short” if the eruption is smaller than 3.2 minutes, and “long” otherwise.

   ```
    faithful$length = ifelse(faithful$eruptions < 3.2,
      "short", "long")
   ```

b. Using the `bwplot` function in the `lattice` package, construct parallel boxplots of the waiting times for the “short” and “long” eruptions.
c. Using the `densityplot` function, construct overlapping density plots of the waiting times of the “short” and “long” eruptions.

4.8 (`ggplot2` graphics). In Exercise 4.7, the waiting times for the Old Faithful geysers were compared for the short and long eruptions where the variable `length` in the `faithful` data frame defines the duration of the eruption.

a. Suppose a data frame `dframe` contains a numeric variable `num.var` and a factor `factor.var`. After the `ggplot2` package has been loaded, then the R commands

   ```
    ggplot(dframe, aes(x = num.var, color = factor.var))
      + geom_density()
   ```

   will construct overlapping density estimates of the variable `num.var` for each value of the factor `factor.var`. Use these commands to construct overlapping density estimates of the waiting times of the geysers with short and long eruptions.
b. With a data frame `dframe` containing a numeric variable `num.var` and a factor `factor.var`, the `ggplot2` syntax

```
ggplot(dframe, aes(y = num.var, x = factor.var))
  + geom_boxplot()
```

will construct parallel boxplots of the variable `num.var` for each value of the factor `factor.var`. Use these commands to construct parallel boxplots of the waiting times of the geysers with short and long eruptions.