mation in the data frame EPIDURALF. Use an area plot as well as a polygon to shade body mass values that are greater than or equal to 40.

Solution: Since the data frame EPIDURALF does not have a variable for body mass index, one is created and stored as BMI. The base function density() is applied to the values in BMI, and the results are stored in the object dens, which records the x and y coordinates for the estimated density curve. Since ggplot2 requires a data frame, the information in dens is stored in a data frame named df.dens for later use when a subset of the values in df.dens (those greater than or equal to 40) are used to create an area plot with the function geom_area(). The area under the density plot for BMI values greater than or equal to 40 is also shaded with geom_polygon; however, care must be taken to ensure the same area is shaded by making sure the points given to the polygon enclose the same area as those provided to the area plot.

R Code 2.66

```
> previous_theme <- theme_set(theme_bw())    # set black-and-white theme
> EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2  # Create BMI
> dens <- density(EPIDURALF$BMI)
> df.dens <- data.frame(x = dens$x, y = dens$y)
> p <- ggplot(data = EPIDURALF, aes(x = BMI)) +
+        geom_density(fill = "gray", alpha = 0.4)
> p1 <- p + geom_area(data = subset(df.dens, x >= 40 &
+        x <= max(EPIDURALF$BMI)), aes(x = x, y = y)) +
+        labs(x = "Body Mass Index", y = "", title = "geom\_area()")
> p1  # Left density plot
> p2 <- p + geom_polygon(data = rbind(c(min(df.dens$x[df.dens$x >= 40]),0),
+        subset(df.dens, x >= 40 & x <= max(EPIDURALF$BMI)),
+        c(max(EPIDURALF$BMI), 0)), aes(x = x, y = y)) +
+        labs(y = "", x = "Body Mass Index",  title = "geom\_polygon()")
> p2  # Right density plot
> theme_set(previous_theme)  # Restore original theme
```

2.9.5.2 Violin Plots

A violin plot is a standard kernel density plot that has been rotated around the x-axis. Like boxplots, violin plots can be used to compare the distribution of a quantitative variable for several levels of a qualitative variable; however, unlike boxplots, violin plots do not hide multi-modality. Consider how the right plot of Figure 2.54 on the facing page (a violin plot) is created from reflecting the left plot (a kernel density) around the x-axis. By default, violin plots in ggplot2 have a vertical orientation with factors appearing on the x-axis.

The default argument for scale= when using geom_violin() is "area", which ensures the area for each side-by-side violin plot is the same. If the number of observations in the side-by-side violin plots are different, use scale = "count" so that the areas of the side-by-side violin plots are proportionally scaled to the number of observations in each violin plot. R Code 2.67 on the next page is used to create Figure 2.55 on page 182. The left side of Figure 2.55 on page 182 creates side-by side violin plots of body mass index values according to the physicians' assessments of ease of palpating a patient using scale = "area". Since the numbers of patients classified as easy, difficult, and impossible to palpate are 207, 114, and 21, respectively, the right plot of Figure 2.55 on page 182 is created using scale = "count" so that the violin plots are scaled proportionally according to the total number of observations.

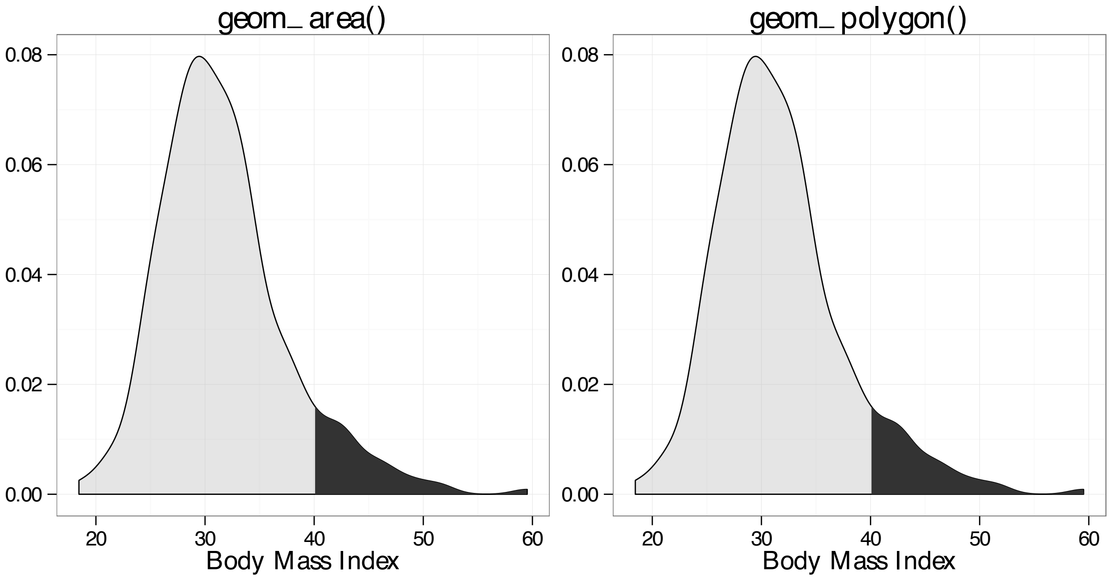

FIGURE 2.53: Density plots that shade BMI values greater than or equal to 40 using two different approaches

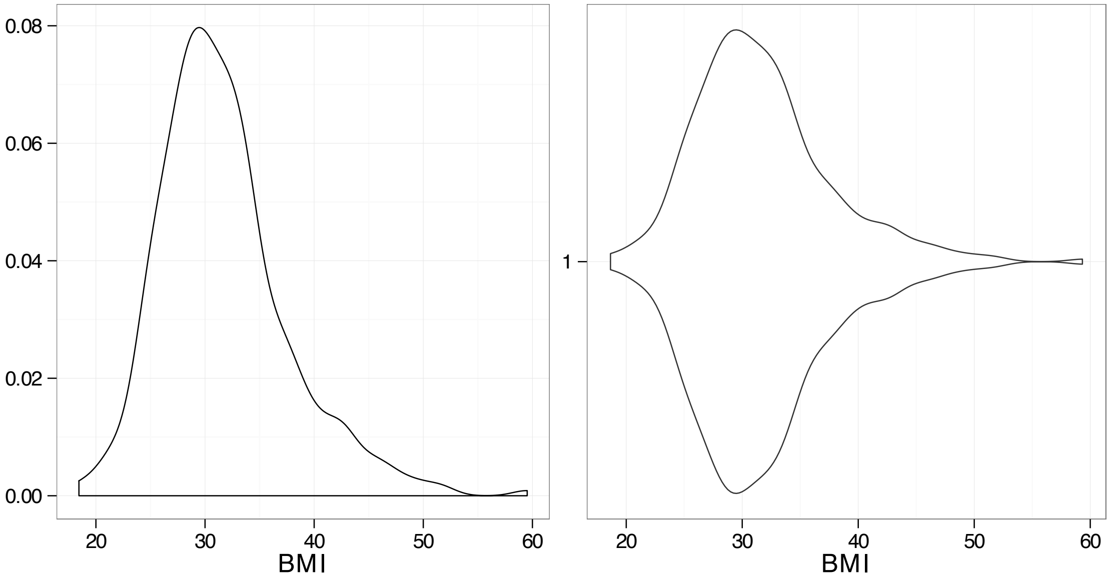

FIGURE 2.54: The left plot is a kernel density plot of the body mass index (BMI) from the data frame EPIDURALF. The right plot shows a violin plot of the body mass index (BMI) from the data frame EPIDURALF.

R Code 2.67

```
> previous_theme <- theme_set(theme_bw())    # set black-and-white theme
> EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2  # Create BMI
> p <- ggplot(data = EPIDURALF, aes(x = ease, y = BMI, fill = ease)) +
+        guides(fill = FALSE) + scale_fill_grey()
> p1 <- p + geom_violin(scale = "area") +
+        labs(title = "Area", x="", y = "Body Mass Index (BMI)")
> p1     # Left area violin plots
> p2 <- p + geom_violin(scale = "count") +
+        labs(title = "Count", x="", y = "Body Mass Index (BMI)")
> p2     # Right count violin plots
> theme_set(previous_theme)  # Restore original theme
```

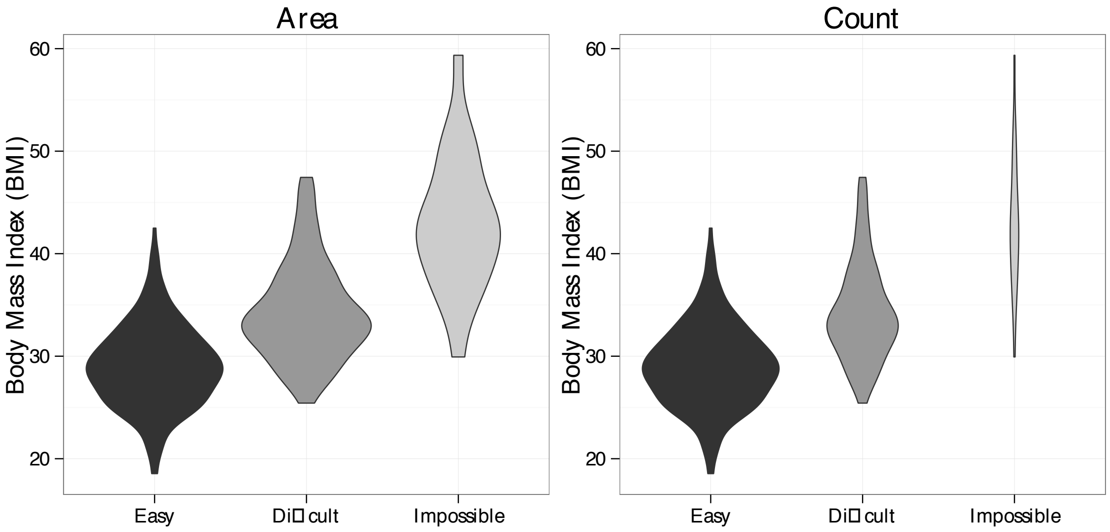

FIGURE 2.55: The left plot shows side-by-side violin plots of body mass index for patients according to the physicians' assessment of ease of palpating their spine using the default scale = "area" argument. The right plot shows side-by-side violin plots of body mass index for patients according to the physicians' assessment of ease of palpating their spine using the argument scale = "count" to create plots scaled proportionally to their number of observations.

R Code 2.68 is used to create Figure 2.56 on the facing page, which shows the relationship between boxplots and count violin plots. The left plot in Figure 2.56 on the next page superimposes the count violin plots with boxplots. The right plot of Figure 2.56 on the facing page adds an additional layer of jittered observations to each count violin plot so the reader can see the relationship between the actual observations and the scaling of the count violin plots.

R Code 2.68

```
> previous_theme <- theme_set(theme_bw())    # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = ease, y = BMI))   # Empty plot
> p1 <- p + geom_violin(scale = "count") +
+        geom_boxplot(aes(fill = ease), width = 0.25, outlier.size = 1.25) +
+        scale_fill_grey() +
+        guides(fill = FALSE) +
+        labs(x="", y = "Body Mass Index (BMI)", title = "Count")
> p1     # Left violin plots/boxplots
> p2 <- p1  + geom_jitter(aes(color = ease), size = 1.25) +
+         scale_color_grey(start = 0.8, end = 0.2) +
+         guides(color = FALSE)
> p2      # Right violin plots/boxplots
> theme_set(previous_theme)                  # Restore original theme
```

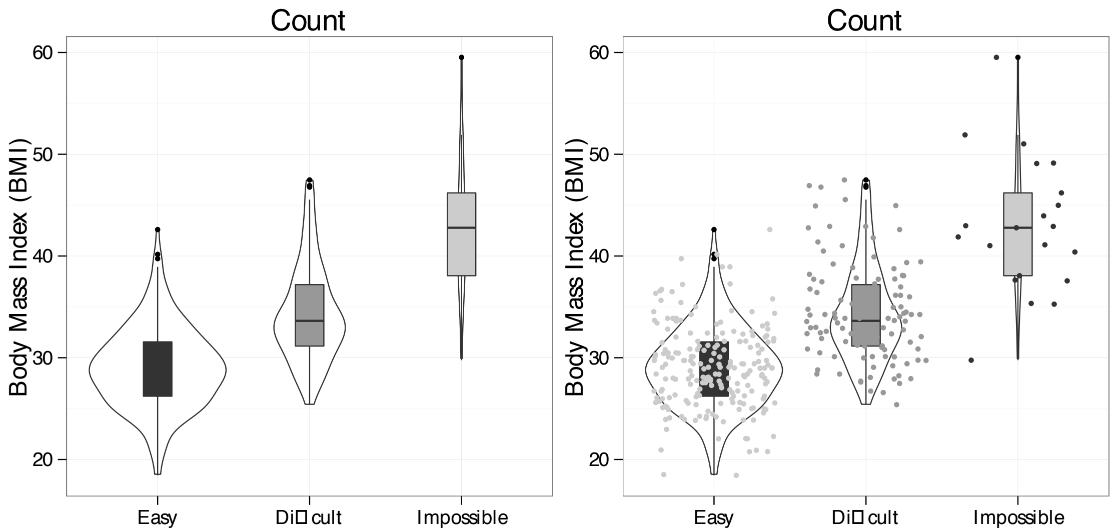

FIGURE 2.56: The left plot shows count violin plots superimposed with boxplots. The right plot adds jittered observations to the left plot.

Strip charts using the base R function stripchart(), often referred to as dot plots, were discussed in Section 2.4.2 on page 105. To create a dotplot with ggplot2, one uses the geom geom_dotplot(). A dotplot similar to Figure 2.7 on page 107, which shows the number of home runs Babe Ruth hit while playing for three different teams, is created with ggplot2 using R Code 2.69 and is shown in the left plot of Figure 2.57 on the following page. Although the dotplot shows the distribution for the number of home runs Babe Ruth hit while playing for three different baseball teams, it does not make use of the year variable. The right plot of Figure 2.57 on the next page shows a scatterplot of home runs hit versus year faceted on team The right plot of Figure 2.57 on the following page shows how Babe Ruth started out hitting very few home runs for the Bos-A, but in 1918 started to hit more home runs per season and was traded to the NY-A where he spent most of his career hitting between 20 and 60 home runs per season. Babe Ruth's home run production started a steady decline in 1930; and in 1934, Babe was traded to the Bos-N for the 1935 season, which was his last.

R Code 2.69

```
> previous_theme <- theme_set(theme_bw())     # set black-and-white theme
> p <- ggplot(data = BABERUTH, aes(x = hr, fill = team))
> p1 <- p + geom_dotplot() +
+    facet_grid(team ~ .) +
+    scale_fill_grey()
> p1                                           # left dotplots
> p <- ggplot(data = BABERUTH, aes(x = year, y = hr, color = team))
```