Graphics in R

Jim Bentley

1 Sample Data

The following code creates a couple of sample data frames that we will use in our examples.

```
> sex = c(rep("Female",12),rep("Male",7))
> mass = c(36.1, 54.6, 48.5, 42.0, 50.6, 42.0, 40.3, 33.1, 42.4, 34.5,
+          51.1, 41.2, 51.9, 46.9, 62, 62.9, 47.4, 48.7, 51.9)
> rate = c(995, 1425, 1396, 1418, 1502, 1256, 1189, 913, 1124, 1052,
+          1347, 1204, 1867, 1439, 1792, 1666, 1362, 1614, 1460)
> bps5.4.9 = data.frame(sex, mass, rate)
> htwt = read.csv(
+ "http://bulldog2.redlands.edu/facultyfolder/jim_bentley/downloads/math111/htwt.csv")
```

Note that the plus signs (+) at the begining of the lines are there to indicate that R is reading from a new line. They should not be entered as part of the code.

We can now check to see if the data frames have been created by entering

```
> ls()
```

```
[1] "bps5.4.9" "htwt"     "mass"     "rate"     "sex"
```

Note that the listing also shows the individual variables that were used to create the data frame. These can be deleted by using rm().

```
> rm("sex","mass","rate")
> ls()
```

```
[1] "bps5.4.9" "htwt"
```

R contains a number of predefined data frames. Some of these will be used in the examples that are presented below.

2 Graphics

R supports a number of different approaches to generating graphics. We will look at standard R graphics, the lattice package, and graphics using the ggplot2 package.

2.1 Standard R Graphics

To use the standard graphics within R we do not need to load any additional packages. A simple scatterplot of the data from BPS5e problem 4.9 (Figure 1) can be created by entering

```
> plot(bps5.4.9$mass,bps5.4.9$rate, xlab="Lean Body Mass (kilograms)",
+      ylab="Metabolic Rate (calories)")
```

A boxplot of the rate variable (Figure 2) can be generated using

```
> boxplot(bps5.4.9$rate, ylab="Metabolic Rate (calories)")
```

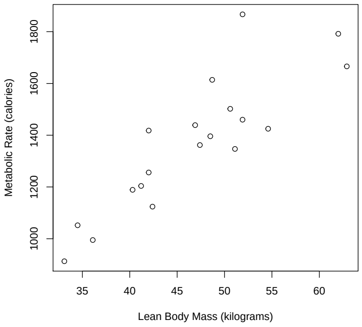

Figure 1: Plot of metabolic rate as a function of lean body mass for the data from BPS5 problem 4.9.

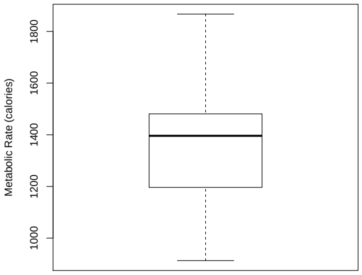

Figure 2: Boxplot of metabolic rate for the data from BPS5 problem 4.9.

A histogram of metabolic rate for the data from BPS5 problem 4.9 (Figure 3) can be generated using

```
> hist(bps5.4.9$rate, xlab="Metabolic Rate (calories)")
```

The corresponding stemplot for the rate data is given by entering

```
> print(stem(bps5.4.9$rate))

  The decimal point is 2 digit(s) to the right of the |

   8 | 1
  10 | 0529
  12 | 0656
  14 | 023460
  16 | 179
  18 | 7

NULL
```

Since this generates a stemplot with too few stems, we may wish to expand the stems a bit. The following function call provides more stems—10 to be exact.

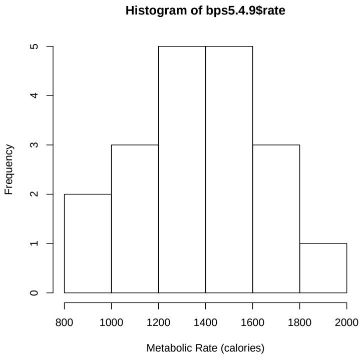

Figure 3: Histogram of the metabolic rates (colories) from BPS5e problem 4.9.

```
> print(stem(bps5.4.9$rate,2))

  The decimal point is 2 digit(s) to the right of the |

   9 | 1
  10 | 05
  11 | 29
  12 | 06
  13 | 56
  14 | 02346
  15 | 0
  16 | 17
  17 | 9
  18 | 7

NULL
```

Of course, it is possible to have too many stems as is shown in the following example.

```
> print(stem(bps5.4.9$rate,5))

  The decimal point is 2 digit(s) to the right of the |

   9 | 1
   9 |
  10 | 0
  10 | 5
  11 | 2
  11 | 9
  12 | 0
  12 | 6
  13 |
  13 | 56
  14 | 0234
  14 | 6
  15 | 0
  15 |
  16 | 1
  16 | 7
  17 |
  17 | 9
  18 |
  18 | 7

NULL
```

2.2 Lattice Graphics

Use of the `lattice` package requires that the package be loaded. Entering

```
> library(lattice)
```

accomplishes this.

A simple scatterplot of the data from BPS5e problem 4.9 (Figure 4) can be created by entering

```
> latticeplot = xyplot(rate~mass, data=bps5.4.9,
+                      xlab="Lean Body Mass (kilograms)", ylab="Metabolic Rate (calories)")
> print(latticeplot)
```

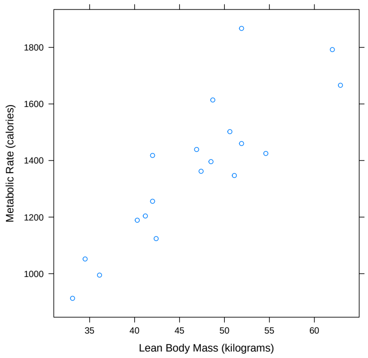

Figure 4: Plot of metabolic rate as a function of lean body mass for the data from BPS5 problem 4.9.

Comparison of sexes can be made by using conditioning (Figure 5).

```
> latticeplot = xyplot(rate~mass|sex, data=bps5.4.9,
+                      xlab="Lean Body Mass (kilograms)", ylab="Metabolic Rate (calories)")
> print(latticeplot)
```

or by the using different symbols for the two groups in overlayed plots (Figure 6).

```
> latticeplot = xyplot(rate~mass, group=sex, pch=c(1,3), data=bps5.4.9,
+                      xlab="Lean Body Mass (kilograms)", ylab="Metabolic Rate (calories)")
> print(latticeplot)
```

A boxplot of the rate variable (Figure 7) can be generated using

```
> latticeplot = bwplot(~rate, data=bps5.4.9,
+                      xlab="Metabolic Rate (calories)")
> print(latticeplot)
```

A boxplot of the rate variable comparing sexes (Figure 8) can be generated using

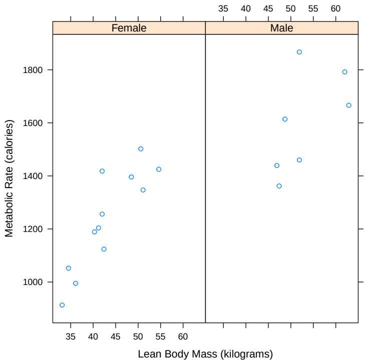

Figure 5: Plot of metabolic rate as a function of lean body mass while controlling for sex for the data from BPS5 problem 4.9.

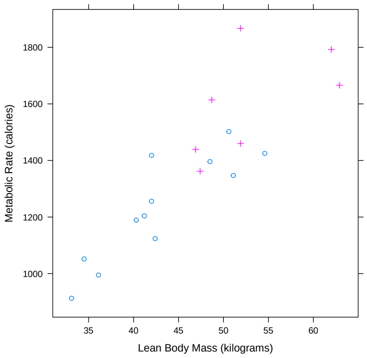

Figure 6: Plot of metabolic rate as a function of lean body mass while controlling for sex for the data from BPS5 problem 4.9.

```
> latticeplot = bwplot(sex~rate, data=bps5.4.9,
+                     ylab="Sex", xlab="Metabolic Rate (calories)")
> print(latticeplot)
```

The `lattice` package includes a few sample data frames. One of these is the `singer` data frame that contains information on various characteristics of some group of singers.

We can create a histogram of the heights of the singers (Figure 9) using

```
> latticeplot = histogram(~height, data=singer)
> print(latticeplot)
```

We can gain additional information by controlling for voice part when creating a histogram of the heights of the singers (Figure 10) using

```
> latticeplot = histogram(~height|voice.part, data=singer)
> print(latticeplot)
```

Similarly, we can look at the distribution of the heights of the singers using density plots. Again, we can gain additional information by controlling for voice part (Figure 11).

```
> latticeplot = densityplot(~height|voice.part,data=singer)
> print(latticeplot)
```

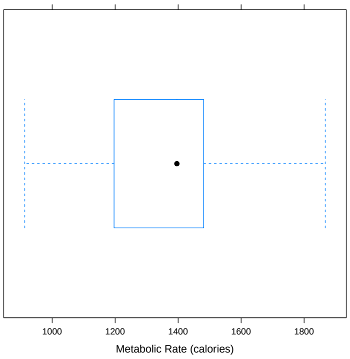

Figure 7: Boxplot of metabolic rate for the data from BPS5 problem 4.9.

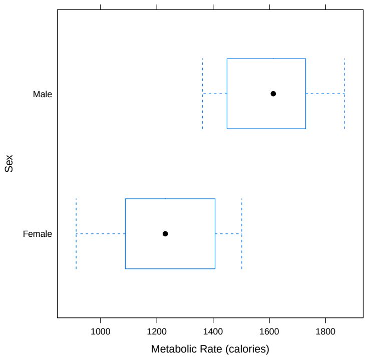

Figure 8: Boxplot of metabolic rate controlling for sex for the data from BPS5 problem 4.9.

One of the nice things about R is that its use of objects means that it is smart about data types. R knows the difference between cardinal (numerical) and categorical (factor) data. The `histogram` function from the `lattice` package will revert to a bargraph when asked to plot a factor variable. Figure 12 shows how this works for the `voice.part` variable.

```
> latticeplot = histogram(~voice.part,data=singer)
> print(latticeplot)
```

Figure 13 is the plot that made the whole idea of trellised graphics famous. The barley data that is presented had been analyzed for years by both the investigators and students. It was not until trellised graphics came along that it was recognized that one of the sites appears to have had its year data swapped.

```
> latticeplot = dotplot(variety ~ yield | site, data = barley, groups = year, pch=c(1,3),
+                       key = simpleKey(levels(barley$year), space = "right"),
+                       xlab = "Barley Yield (bushels/acre) ",
+                       aspect=0.5, layout = c(1,6), ylab=NULL)
> print(latticeplot)
```

2.3 GGPLOT2 Graphics

Use of the GGPLOT2 package requires that the package be loaded. Entering

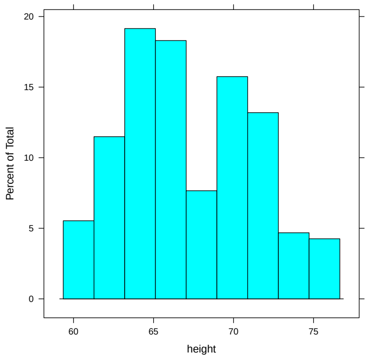

Figure 9: Histogram of the heights of singers in the `singer` data frame.

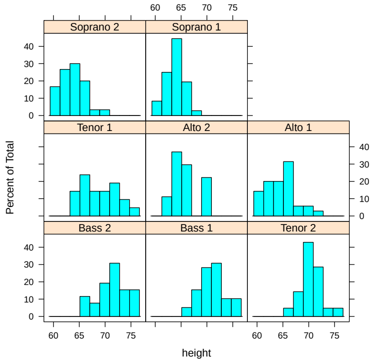

Figure 10: Histogram of the heights of singers in the `singer` data frame controlling for voice part.

```
> library(ggplot2)
```

accomplishes this. The structure of ggplot is quite different from standard R and lattice graphics. To generate a boxplot of metabolic rate that allows a comparison by sex (Figure 14) one enters the following commands.

```
> bw = ggplot(bps5.4.9,aes(sex,rate))
> bw = bw + ylab("Metabolic Rate (calories)") + xlab("Sex")
> bw = bw + geom_boxplot() + coord_flip()
> print(bw)
```

A histogram of metabolic rate (Figure 15) is made by entering the following code.

```
> plt = ggplot(bps5.4.9, aes(x=rate))
> plt = plt + xlab("Metabolic Rate  (calories)")
> plt = plt + geom_histogram(binwidth=200)
> print(plt)
```

The sentax for a bar chart is similar to that of a histogram. Figure 16 shows a bar chart of the `sex` variable from the BPS5e problem 4.9 data.

```
> plt = ggplot(bps5.4.9, aes(x=sex))
> plt = plt + geom_bar()
> plt = plt + xlab("Sex")
> print(plt)
```

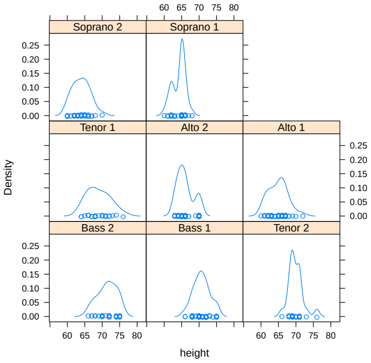

Figure 11: Density plot of the heights of singers in the `singer` data frame controlling for voice part.

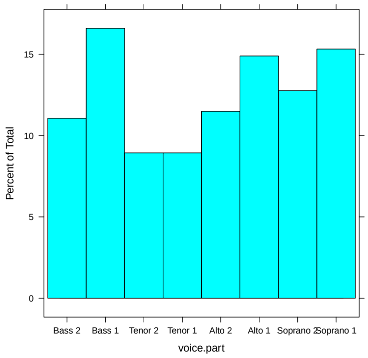

Figure 12: Bargraph generated using `histogram` on the factor variable `voice.part` from the `singer` data frame.

GGPLOT2 also provides scatterplots that can be enhanced with things like LOESS smooths (Figure 17).

```
> plt = ggplot(bps5.4.9, aes(mass, rate, shape=sex, linetype=sex))
> plt = plt + xlab("Mass (kilograms)") + ylab("Metabolic Rate (calories)")
> plt = plt + geom_point(size=3) + stat_smooth(span=0.8, colour="black", lwd=0.25)
> print(plt)
```

As with the `lattice` package, it is possible to create separate plots for each of the sexes by using (Figure 18).

```
> plt = ggplot(bps5.4.9, aes(mass, rate)) + facet_grid(sex~.)
> plt = plt + xlab("Mass (kilograms)") + ylab("Metabolic Rate (calories)")
> plt = plt + geom_point(size=3) + stat_smooth(span=0.8, colour="black", lwd=0.25)  + theme_bw()
> print(plt)
```

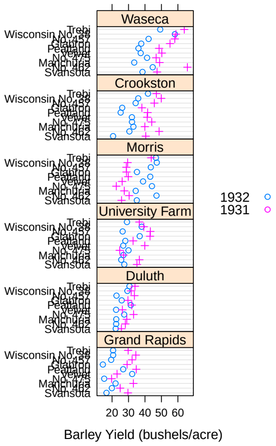

Figure 13: Density plot of the `barley` data showing the reversal of year for one of the study sites.

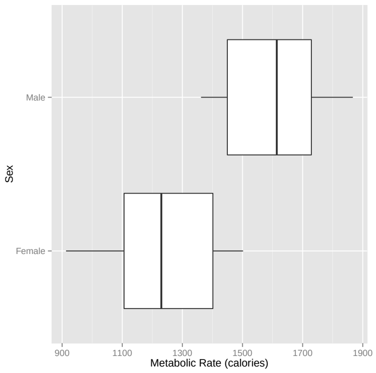

Figure 14: Boxplots of metabolic rate by sex for the data from BPS5 problem 4.9.

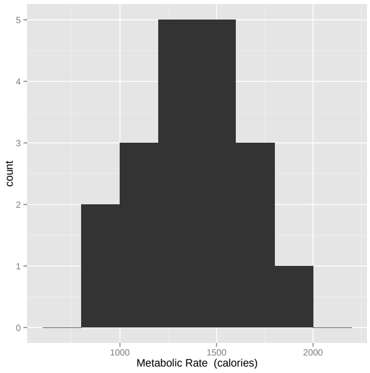

Figure 15: Histogram of metabolic rate for the data from BPS5 problem 4.9.

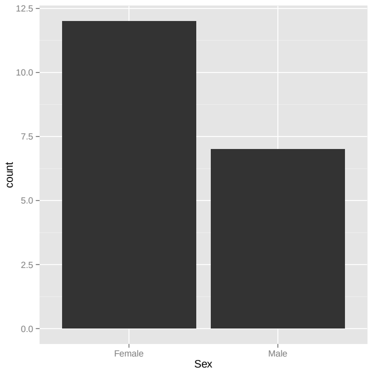

Figure 16: Bar chart of sex for the data from BPS5 problem 4.9.

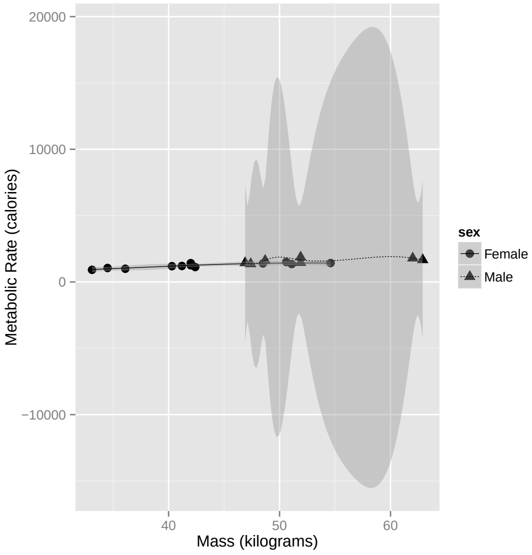

Figure 17: LOESS fits with approximate 95% confidence bounds for all data by sex.

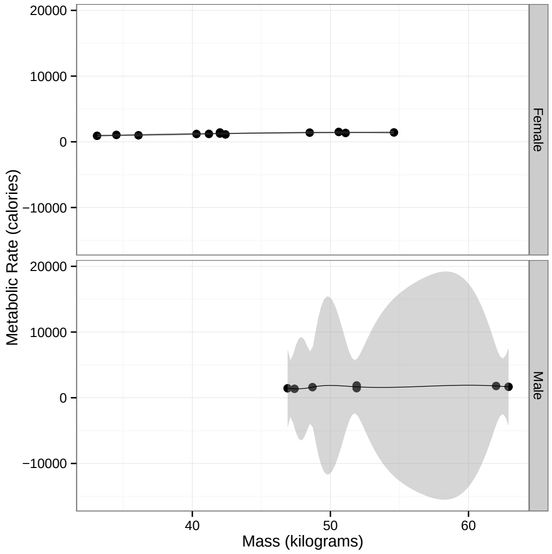

Figure 18: LOESS fits with approximate 95% confidence bounds for all data by sex.