Graphics in R

JuJu and Oliver

## Sample Data

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

### Create a dataframe the hard way  
 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)  
   
 ### Read dataframes from files on a server  
 htwt = read.csv( "http://facweb1.redlands.edu/fac/jim\_bentley/downloads/math111/htwt.csv")  
 Titanic = read.csv("http://facweb1.redlands.edu/fac/jim\_bentley/downloads/math111/titanic.csv")  
 ### Change character strings to factor variables  
 Titanic <- within(Titanic, {Survival <- factor(SURVIVED, labels=c("Died","Survived"))  
})  
 Titanic <- within(Titanic, {Age.Group<-factor(AGE,labels=c("Child","Adult"))})  
 Titanic <- within(Titanic,{Sex <- factor(SEX, labels=c("Female","Male"))})  
 Titanic <- within(Titanic,{Class<-factor(CLASS,labels=c("Crew","First","Second","Steerage"))})  
 head(Titanic)

## CLASS AGE SEX SURVIVED Survival Age.Group Sex Class  
## 1 1 1 1 1 Survived Adult Male First  
## 2 1 1 1 1 Survived Adult Male First  
## 3 1 1 1 1 Survived Adult Male First  
## 4 1 1 1 1 Survived Adult Male First  
## 5 1 1 1 1 Survived Adult Male First  
## 6 1 1 1 1 Survived Adult Male First

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

ls()

## [1] "bps5.4.9" "htwt" "mass" "rate" "sex" "Titanic"

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" "Titanic"

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

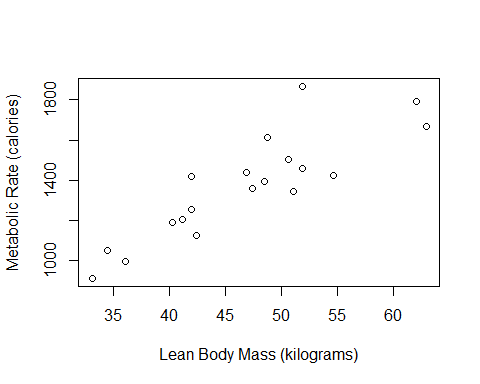
## Graphics

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.

### Standard R Graphics

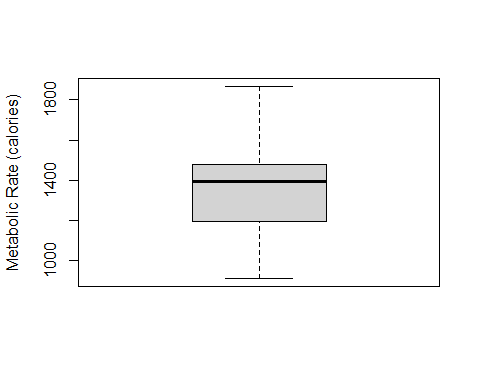
To use the standard graphics within R we do not need to load any additional packages. A simple scatterplot of the data from BPS5e problem 4.9 can be created by entering

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



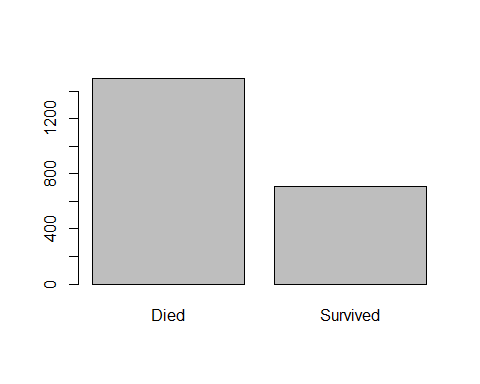
A boxplot of the rate variable can be generated using

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



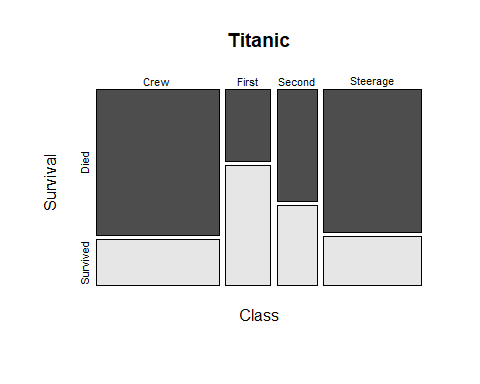
A barchart of the survival rate in the Titanic data can be generated using

barplot(xtabs(~Survival,data=Titanic))



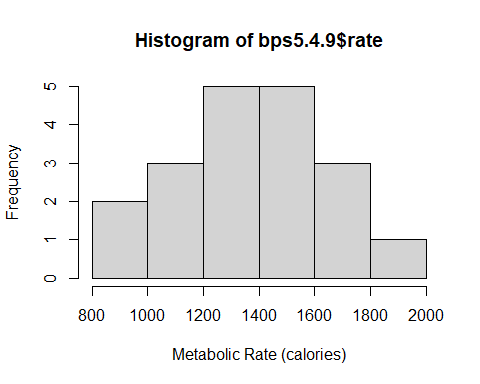
This plot indicates the marginal survival rates that are visible in the mosaic plot of Survival as a function of Class. The mosaic plot can be generated by entering

mosaicplot(~ Class + Survival, data=Titanic, color=TRUE)



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

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



The corresponding stemplot for the rate data is given by entering

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

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.

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

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

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

### Lattice Graphics

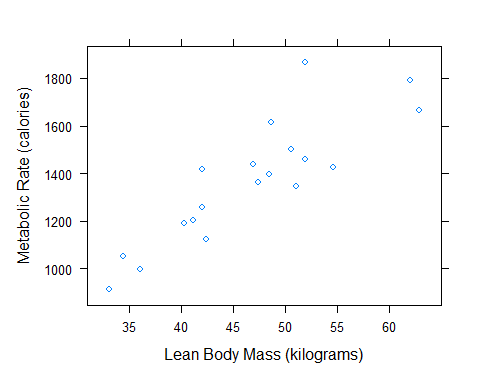
Use of the ***lattice*** package requires that the package be loaded. Entering

p\_load(lattice)

accomplishes this.

A simple scatterplot of the data from BPS problem 4.9

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



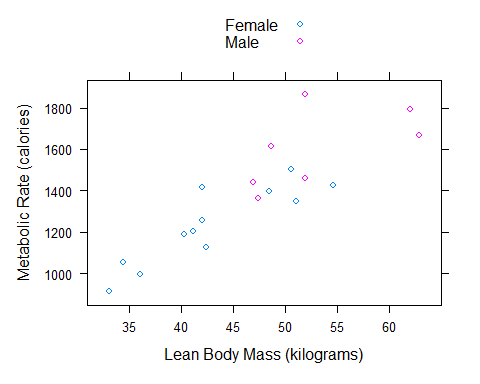
Comparison of sexes can be made by using conditioning

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



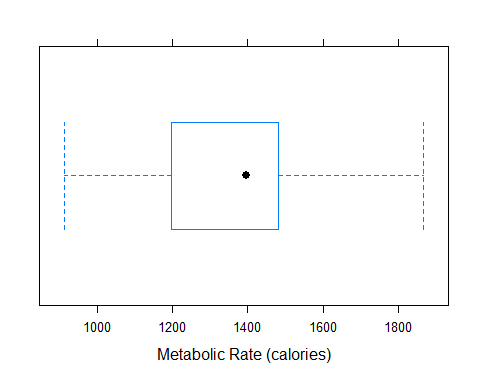
or by the using different symbols for the two groups in overlayed plots

xyplot(rate~mass, group=sex,, data=bps5.4.9, xlab="Lean Body Mass (kilograms)", ylab="Metabolic Rate (calories)", auto.key=TRUE)



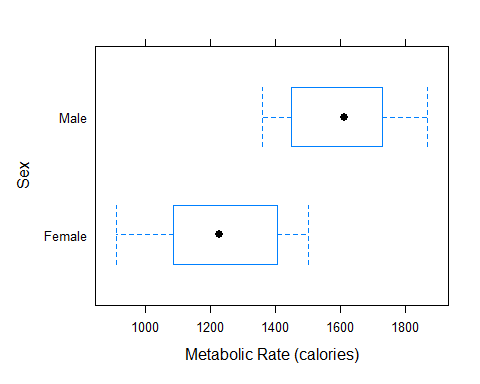
A boxplot of the rate variable can be generated using

bwplot(~rate, data=bps5.4.9, xlab="Metabolic Rate (calories)")



A boxplot of the rate variable comparing sexes can be generated using

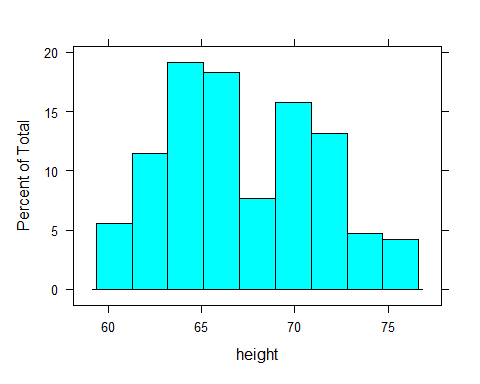
bwplot(sex~rate, data=bps5.4.9, ylab="Sex", xlab="Metabolic Rate (calories)")



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.

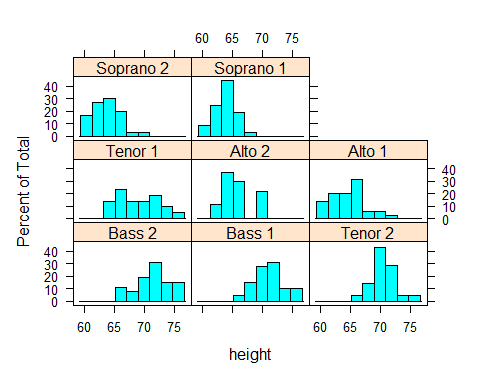
We can create a histogram of the heights of the singers using

histogram(~height, data=singer)



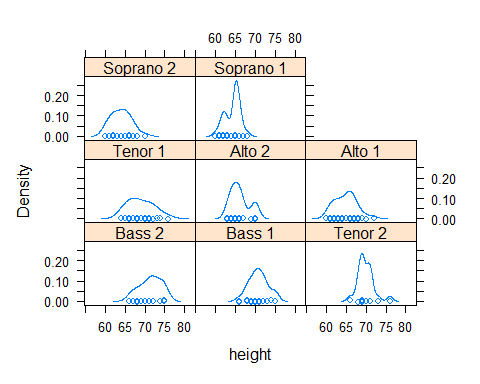
We can gain additional information by controlling for voice part when creating a histogram of the heights of the singers using

histogram(~height|voice.part, data=singer)



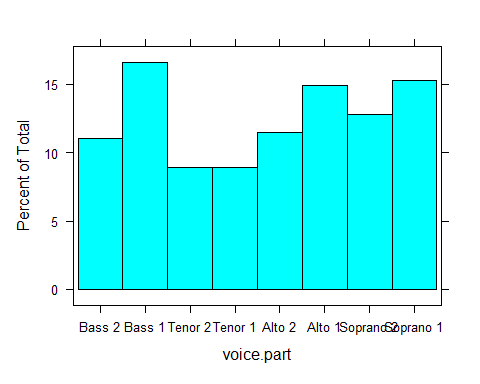
Similarly, we can look at the distribution of the heights of the singers using density plots. Again, we can gain additional information by controlling for voice part

densityplot(~height|voice.part,data=singer)



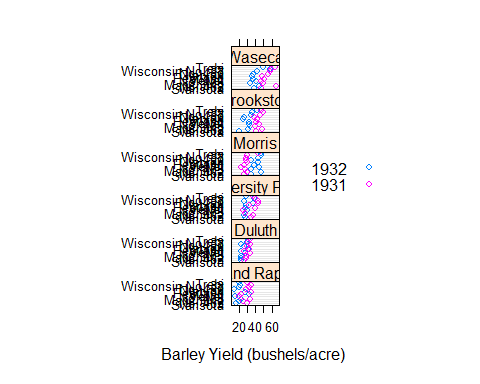
One of the nice things about R is that its use of objects means that it is smart about data types. R knows the difference between cardinal (numerical) and categorical (factor) data. The ***histogram*** function from the ***lattice*** package will revert to a bargraph when asked to plot a factor variable.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

histogram(~voice.part,data=singer)



Below 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.

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



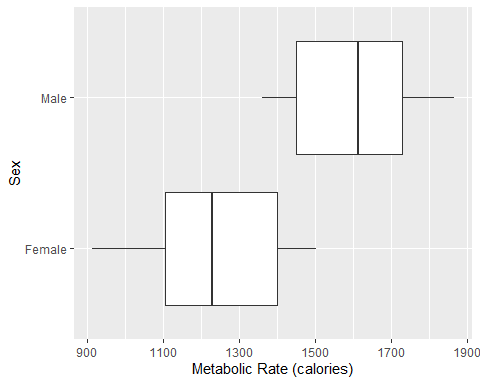
### GGPLOT2 Graphics

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

p\_load(ggplot2)

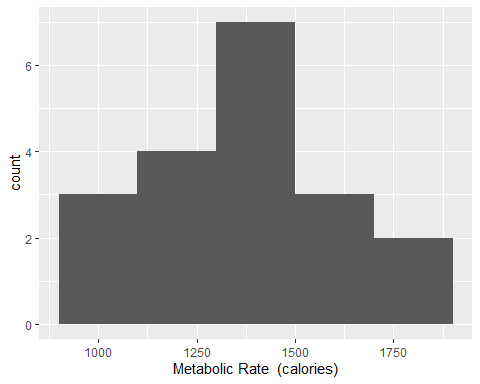
accomplishes this. The structure of ggplot is quite different from standard R and lattice graphics. To generate a boxplot of metabolic rate that allows a comparison by sex one enters the following commands.

bw = ggplot(bps5.4.9,aes(sex,rate))  
 bw = bw + ylab("Metabolic Rate (calories)") + xlab("Sex")  
 bw = bw + geom\_boxplot() + coord\_flip()  
 bw



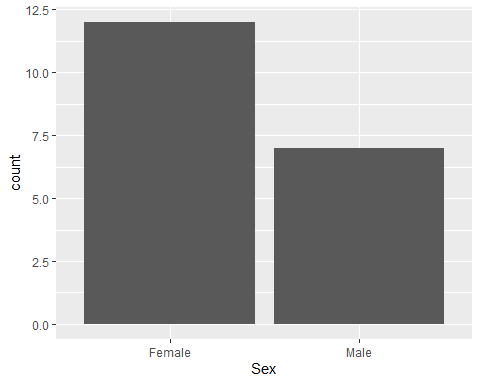
A histogram of metabolic rate is made by entering the following code.

plt = ggplot(bps5.4.9, aes(x=rate))  
 plt = plt + xlab("Metabolic Rate (calories)")   
 plt = plt + geom\_histogram(binwidth=200)  
 plt



The sentax for a bar chart is similar to that of a histogram. The figure below shows a bar chart of the ***sex*** variable from the BPS Problem 4.9 data.

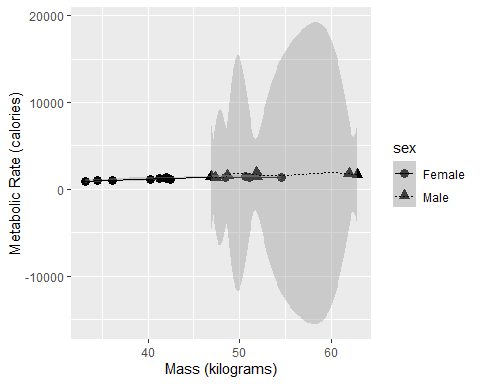
plt = ggplot(bps5.4.9, aes(x=sex))  
 plt = plt + geom\_bar()  
 plt = plt + xlab("Sex")   
 plt



**GGPLOT2** also provides scatterplots that can be enhanced with things like LOESS smooths

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) + geom\_smooth(method="loess", span=0.8, colour="black", lwd=0.25)   
 plt

## `geom\_smooth()` using formula 'y ~ x'



As with the ***lattice*** package, it is possible to create separate plots for each of the sexes by using

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) + geom\_smooth(method="loess", span=0.8, colour="blue")   
 plt

## `geom\_smooth()` using formula 'y ~ x'

