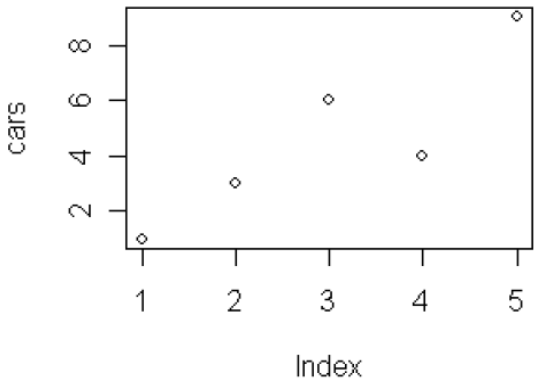
Producing Simple Graphs with R

The following is an introduction for producing simple graphs with the R Programming Language. Each example builds on the previous one. The areas in **bold** indicate new text that was added to the previous example. The graph produced by each example is shown on the right.

Line Charts

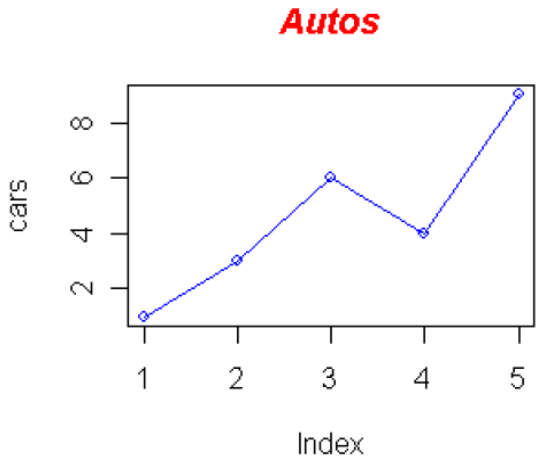
First we'll produce a very simple graph using the values in the car vector:

```
# Define the cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)

# Graph the cars vector with all defaults
plot(cars)
```



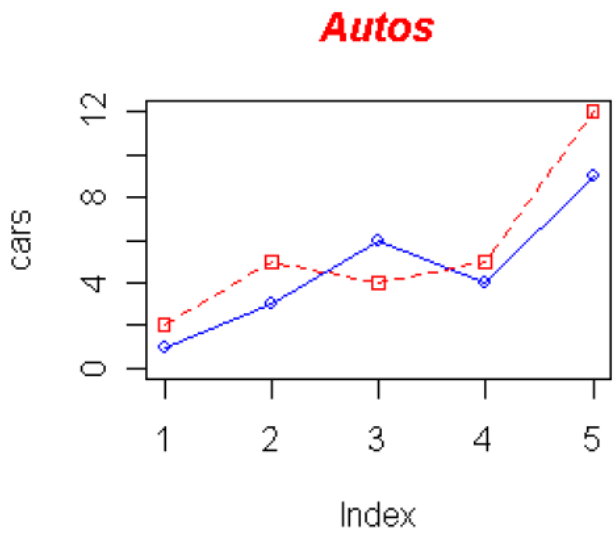
Let's add a title, a line to connect the points, and some color:

```
# Define the cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)

# Graph cars using blue points overlayed by
a line
plot(cars, type="o", col="blue")

# Create a title with a red, bold/italic
font
title(main="Autos", col.main="red",
font.main=4)
```



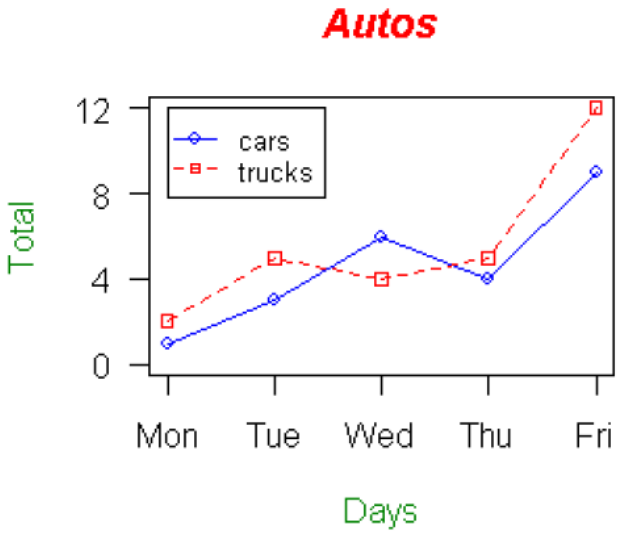
Now let's add a red line for trucks and specify the y-axis range directly so it will be large enough to fit the truck data:

```
# Define 2 vectors
cars <- c(1, 3, 6, 4, 9)
trucks <- c(2, 5, 4, 5, 12)

# Graph cars using a y axis that ranges from
0 to 12
plot(cars, type="o", col="blue",
ylim=c(0,12))

# Graph trucks with red dashed line and
square points
lines(trucks, type="o", pch=22, lty=2,
col="red")

# Create a title with a red, bold/italic
font
title(main="Autos", col.main="red",
font.main=4)
```



Next let's change the axes labels to match our data and add a legend. We'll also compute the y-axis values using the max function so any changes to our data will be automatically reflected in our graph.

```
# Define 2 vectors
cars <- c(1, 3, 6, 4, 9)
trucks <- c(2, 5, 4, 5, 12)

# Calculate range from 0 to max value of
cars and trucks
g_range <- range(0, cars, trucks)

# Graph autos using y axis that ranges from
0 to max
# value in cars or trucks vector.  Turn off
axes and
# annotations (axis labels) so we can
specify them ourself
plot(cars, type="o", col="blue",
ylim=g_range,
   axes=FALSE, ann=FALSE)

# Make x axis using Mon-Fri labels
axis(1, at=1:5,
lab=c("Mon","Tue","Wed","Thu","Fri"))

# Make y axis with horizontal labels that
display ticks at
# every 4 marks. 4*0:g_range[2] is
equivalent to c(0,4,8,12).
axis(2, las=1, at=4*0:g_range[2])

# Create box around plot
box()
```



```
# Create a title with a red, bold/italic
font
title(main="Autos", col.main="red",
font.main=4)

# Label the x and y axes with dark green
text
title(xlab="Days", col.lab=rgb(0,0.5,0))
title(ylab="Total", col.lab=rgb(0,0.5,0))

# Create a legend at (1, g_range[2]) that is
slightly smaller
# (cex) and uses the same line colors and
points used by
# the actual plots
legend(1, g_range[2], c("cars","trucks"),
cex=0.8,
   col=c("blue","red"), pch=21:22, lty=1:2);
```

Now let's create a data frame with 3 columns: cars, trucks, suvs by binding them together .

suvs<-c(4,4,6,6,12)

autos_data<-data.frame(cbind(cars,trucks,suvs))

Type autos_data, you will see this:

```
> autos_data
  cars trucks suvs
1    1      2    4
2    3      5    4
3    6      4    6
4    4      5    6
5    9     12   16
```

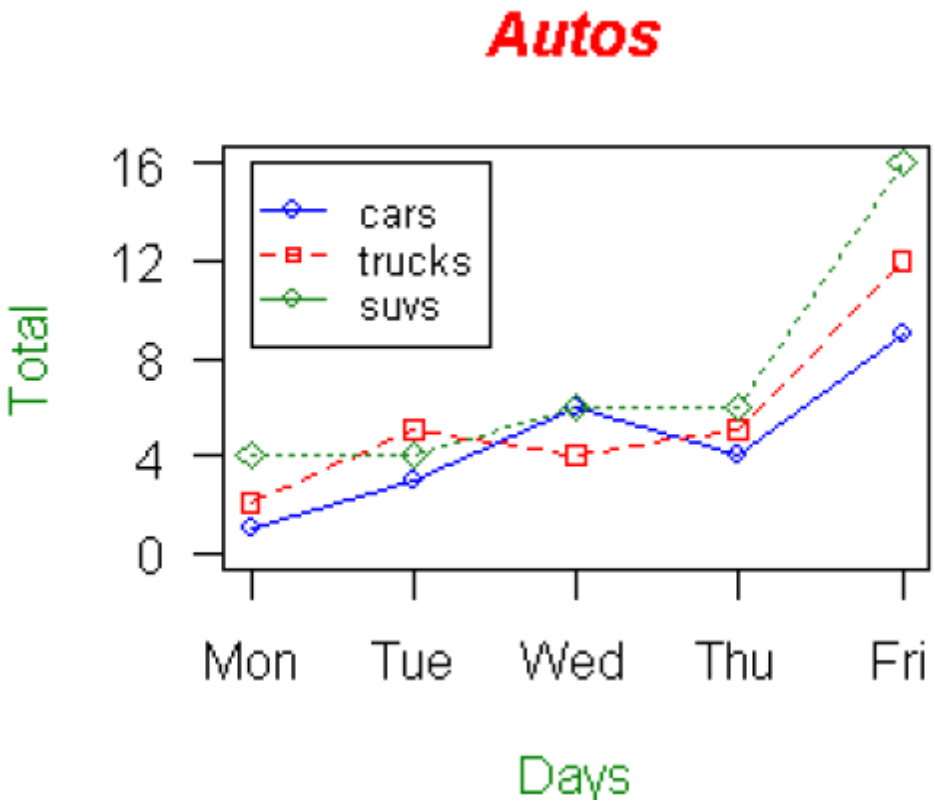
We'll also use a vector for storing the colors to be used in our graph so if we want to change the colors later on, there's only one place in the file that needs to be modified.

```
# Define colors to be used for cars, trucks,
suvs
plot_colors <- c("blue","red","forestgreen")
```

```
# Compute the largest y value used in the
data (or we could
# just use range again)
max_y <- max(autos_data)

# Graph autos using y axis that ranges from
0 to max_y.
# Turn off axes and annotations (axis
labels) so we can
# specify them ourself
plot(autos_data$cars, type="o",
col=plot_colors[1],
   ylim=c(0,max_y), axes=FALSE, ann=FALSE)

# Make x axis using Mon-Fri labels
axis(1, at=1:5, lab=c("Mon", "Tue", "Wed",
"Thu", "Fri"))

# Make y axis with horizontal labels that
display ticks at
# every 4 marks. 4*0:max_y is equivalent to
c(0,4,8,12).
axis(2, las=1, at=4*0:max_y)

 # Create box around plot
 box()

 # Graph trucks with red dashed line and
 square points
 lines(autos_data$trucks, type="o", pch=22,
 lty=2,
    col=plot_colors[2])

 # Graph suvs with green dotted line and
 diamond points
 lines(autos_data$suvs, type="o", pch=23,
 lty=3,
    col=plot_colors[3])
```

```
# Create a title with a red, bold/italic
font
title(main="Autos", col.main="red",

 font.main=4)

# Label the x and y axes with dark green
text
title(xlab= "Days", col.lab=rgb(0,0.5,0))
title(ylab= "Total", col.lab=rgb(0,0.5,0))

# Create a legend at (1, max_y) that is
slightly smaller
# (cex) and uses the same line colors and
points used by
# the actual plots
legend(1, max_y, names(autos_data), cex=0.8,
col=plot_colors,
   pch=21:23, lty=1:3);
```

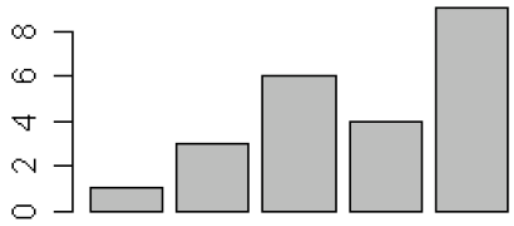
Your plot should look like this one



Bar Charts

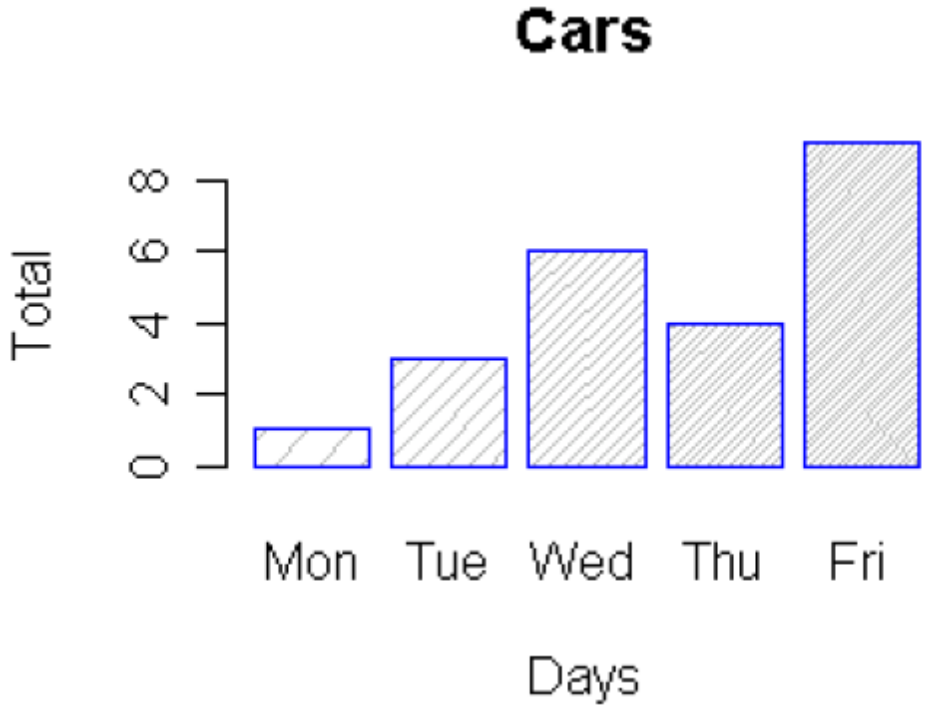
Let's start with a simple bar chart graphing the cars vector:

```
# Define the cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)

# Graph cars
barplot(cars)
```



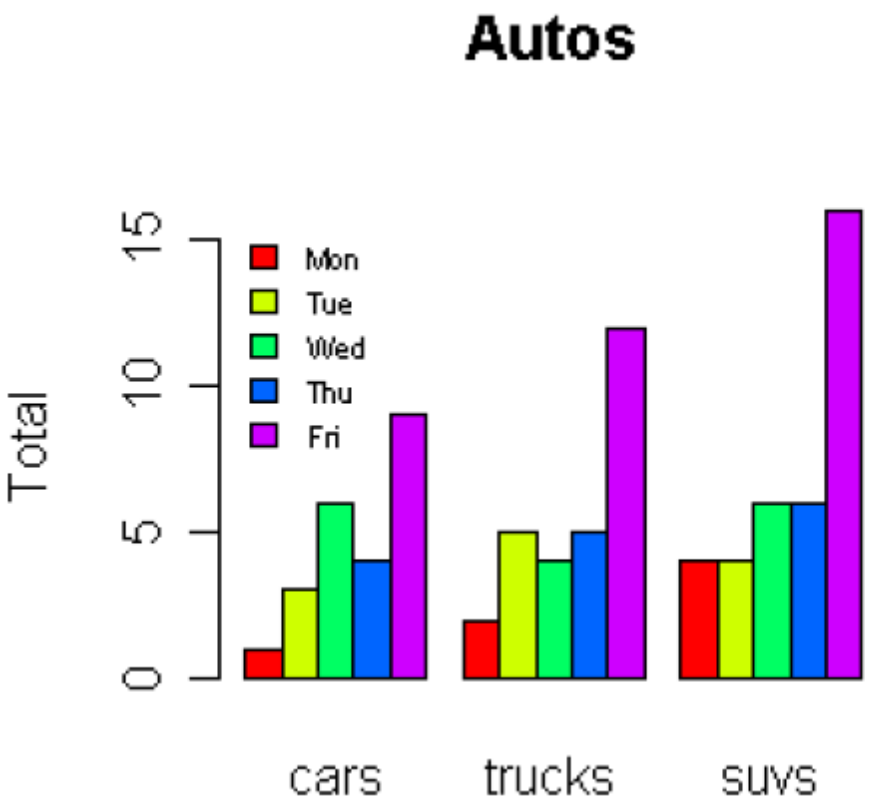
Let's now read data from autos_data frame, add labels, blue borders around the bars, and density lines.

```
# Graph cars with specified labels for axes.
Use blue
# borders and diagnal lines in bars.
barplot(autos_data$cars, main="Cars",
xlab="Days",
   ylab="Total",
names.arg=c("Mon","Tue","Wed","Thu","Fri"),
   border="blue", density=c(10,20,30,40,50))
```



Now let's graph the total number of autos per day using some color and show a legend:

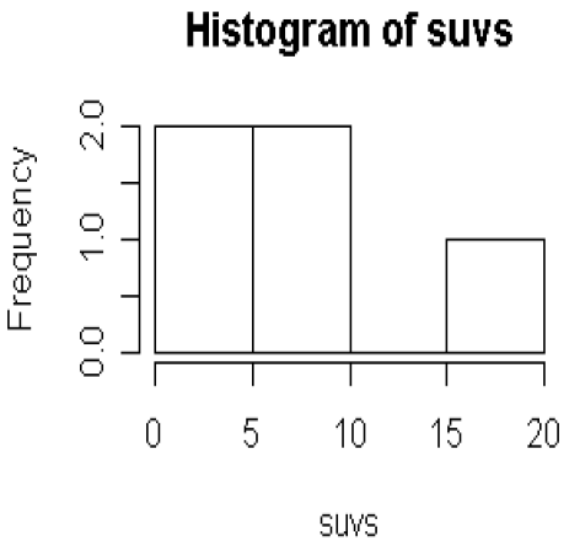
```
# Graph autos with adjacent bars using
rainbow colors
barplot(as.matrix(autos_data), main="Autos",
ylab= "Total",
   beside=TRUE, col=rainbow(5))

# Place the legend at the top-left corner
with no frame
# using rainbow colors
legend("topleft",
c("Mon","Tue","Wed","Thu","Fri"), cex=0.6,
   bty="n", fill=rainbow(5));
```



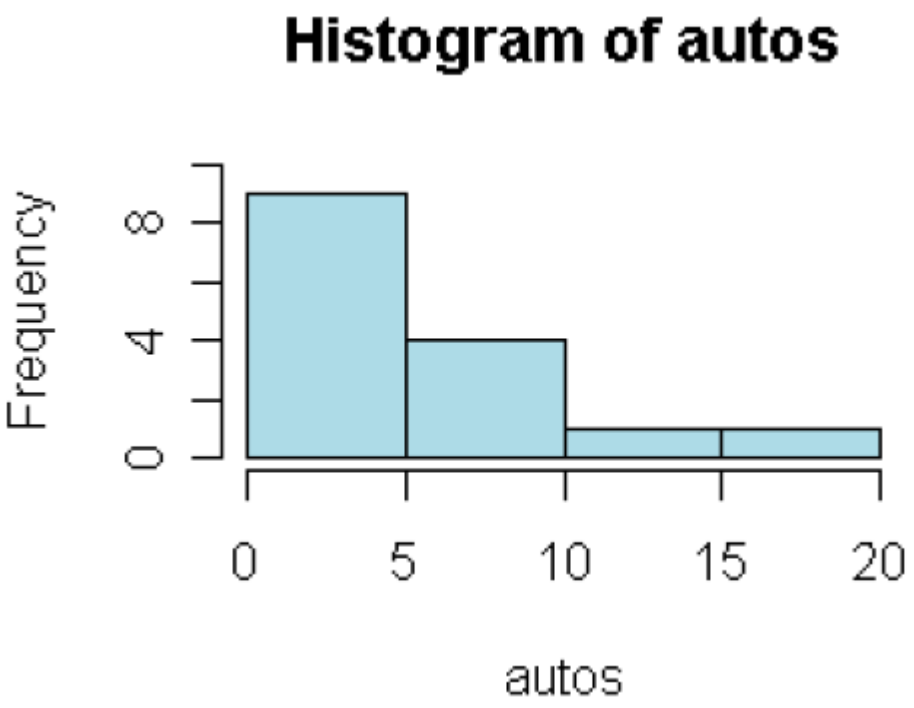
Histograms

Let's start with a simple histogram graphing the distribution of the suvs vector:

```
# Define the suvs vector with 5 values
suvs <- c(4,4,6,6,16)

# Create a histogram for suvs
hist(suvs)
```



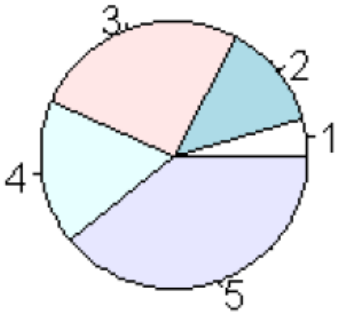

```
# Concatenate the three vectors
autos <- c(autos_data$cars,
autos_data$trucks,
   autos_data$suvs)

# Create a histogram for autos in light blue
with the y axis
# ranging from 0-10
hist(autos, col="lightblue", ylim=c(0,10))
```



Pie Charts

Let's start with a simple pie chart graphing the cars vector:

```
# Define cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)

# Create a pie chart for cars
pie(cars)
```

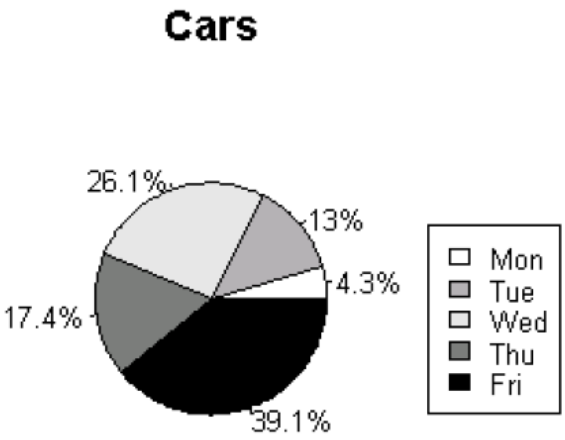


Now let's add a heading, change the colors, and define our own labels:

```
# Define cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)

# Create a pie chart with defined heading
and
# custom colors and labels
pie(cars, main="Cars",
col=rainbow(length(cars)),
   labels=c("Mon","Tue","Wed","Thu","Fri"))
```

Now let's change the colors, label using percentages, and create a legend:

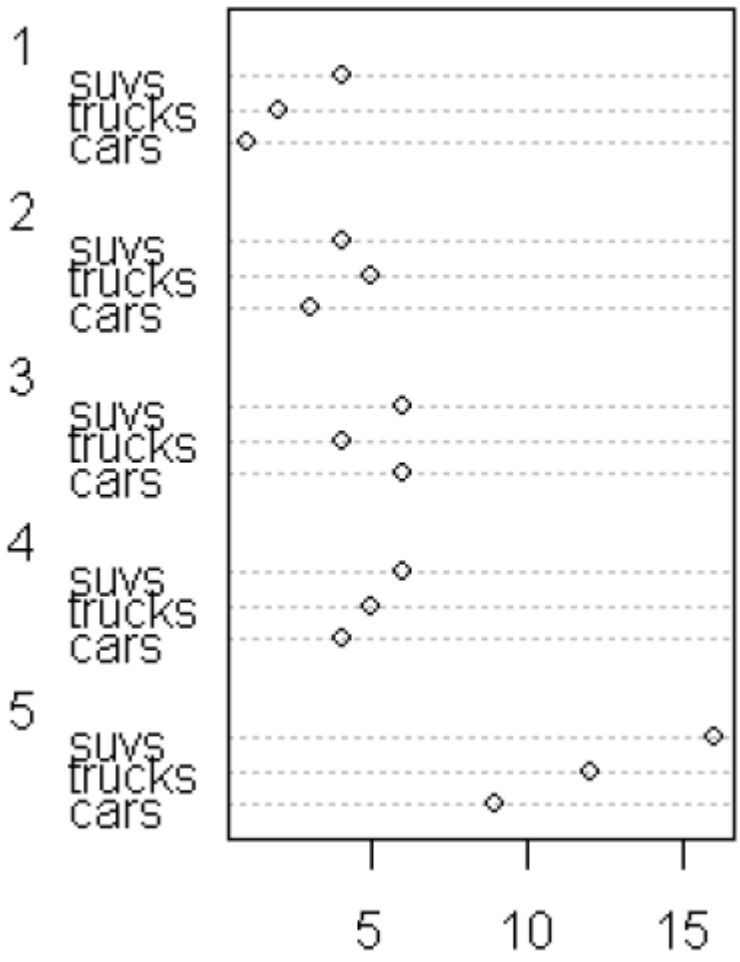
```
# Define cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)

# Define some colors ideal for black & white
print
colors <-
c("white","grey70","grey90","grey50","black"
)

# Calculate the percentage for each day,
rounded to one
# decimal place
car_labels <- round(cars/sum(cars) * 100, 1)

# Concatenate a '%' char after each value
car_labels <- paste(car_labels, "%", sep="")

# Create a pie chart with defined heading
and custom colors
# and labels
pie(cars, main="Cars", col=colors,
labels=car_labels,
   cex=0.8)

# Create a legend at the right
legend(1.5, 0.5,
c("Mon","Tue","Wed","Thu","Fri"), cex=0.8,
   fill=colors)
```



Dotcharts

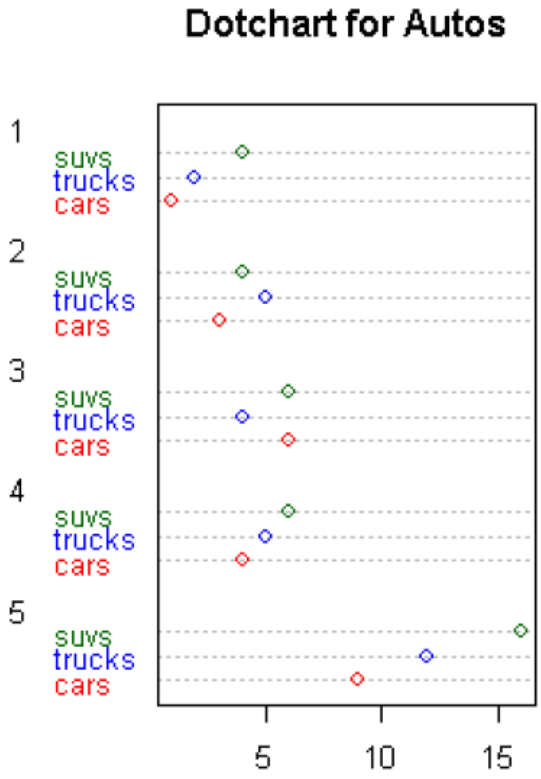
Let's start with a simple dotchart graphing the autos data:

```
# Create a dotchart for autos
dotchart(t(autos_data))
```



Let's make the dotchart a little more colorful:

```
# Read values from tab-delimited autos.dat
autos_data <- read.table("C:/R/autos.dat",
header=T, sep="\t")

# Create a colored dotchart for autos with
smaller labels
dotchart(t(autos_data),
color=c("red","blue","darkgreen"),
   main="Dotchart for Autos", cex=0.8)
```



This example shows all 25 symbols that you can use to produce points in your graphs:

```
# Make an empty chart
plot(1, 1, xlim=c(1,5.5), ylim=c(0,7),
type="n", ann=FALSE)

# Plot digits 0-4 with increasing size and
color
text(1:5, rep(6,5), labels=c(0:4), cex=1:5,
col=1:5)

# Plot symbols 0-4 with increasing size and
color
points(1:5, rep(5,5), cex=1:5, col=1:5,
pch=0:4)
text((1:5)+0.4, rep(5,5), cex=0.6, (0:4))

# Plot symbols 5-9 with labels
points(1:5, rep(4,5), cex=2, pch=(5:9))
text((1:5)+0.4, rep(4,5), cex=0.6, (5:9))

# Plot symbols 10-14 with labels
points(1:5, rep(3,5), cex=2, pch=(10:14))
text((1:5)+0.4, rep(3,5), cex=0.6, (10:14))

# Plot symbols 15-19 with labels
points(1:5, rep(2,5), cex=2, pch=(15:19))
text((1:5)+0.4, rep(2,5), cex=0.6, (15:19))

# Plot symbols 20-25 with labels
points((1:6)*0.8+0.2, rep(1,6), cex=2,
pch=(20:25))
text((1:6)*0.8+0.5, rep(1,6), cex=0.6,
(20:25))
```