Plotting examples in R base graphics and ggplot

Katie Gostic

February 23, 2016

```
## Loading required package: knitr
```

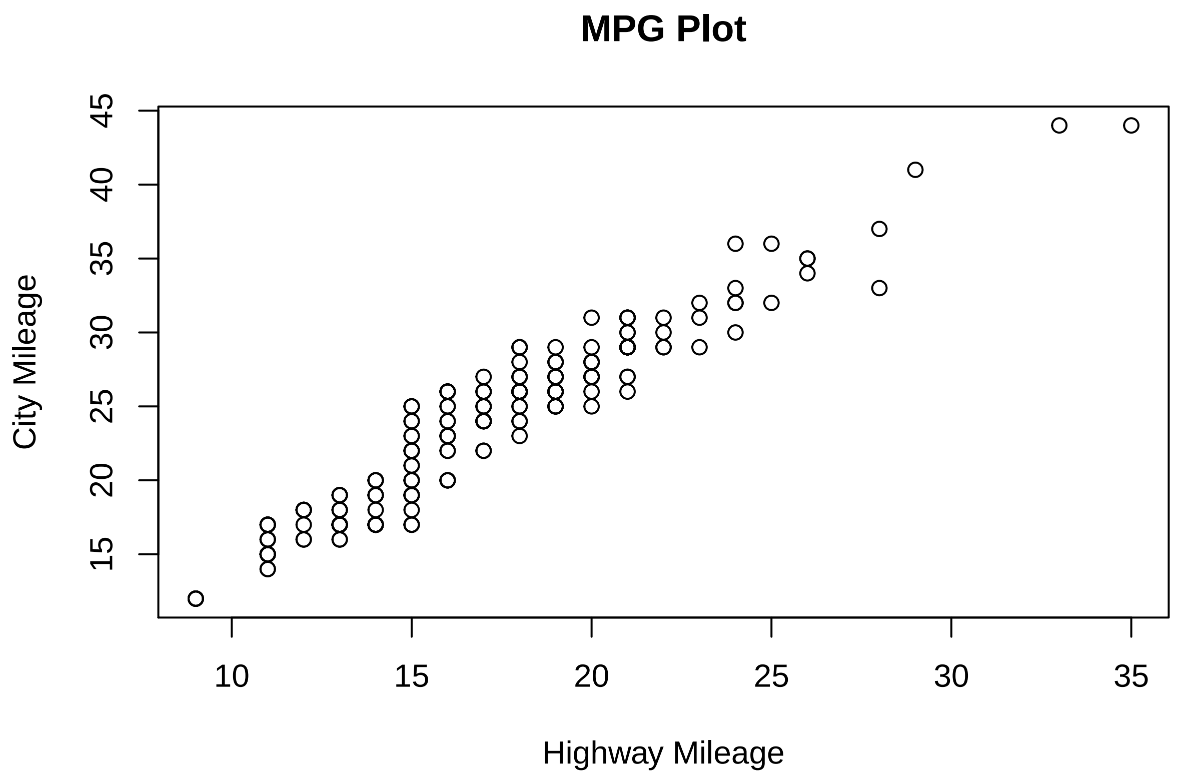
Load packages and data

```
#Load the ggplot package
library(ggplot2)

data(mpg) #Load built-in data set, mpg
?cars #Inspect your data set and get to know the variables
head(mpg)
```

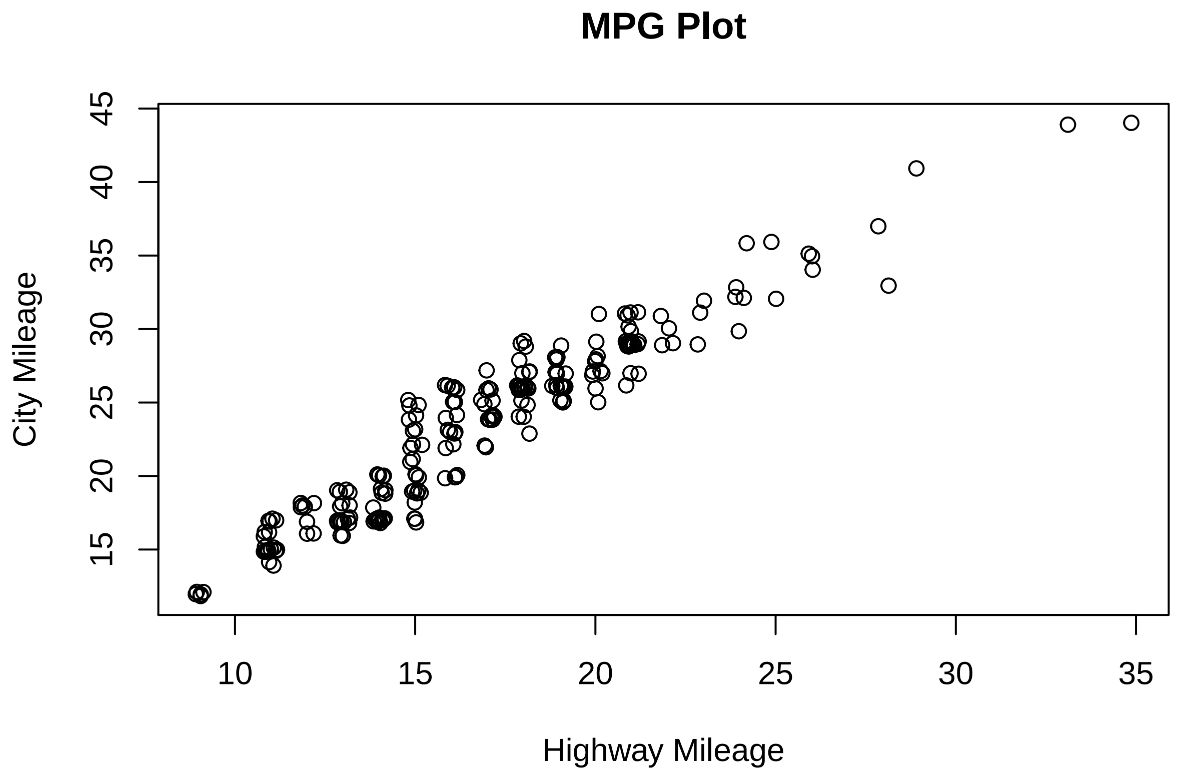
```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
```

Section 1. Examples in R base graphics using the mpg dataset

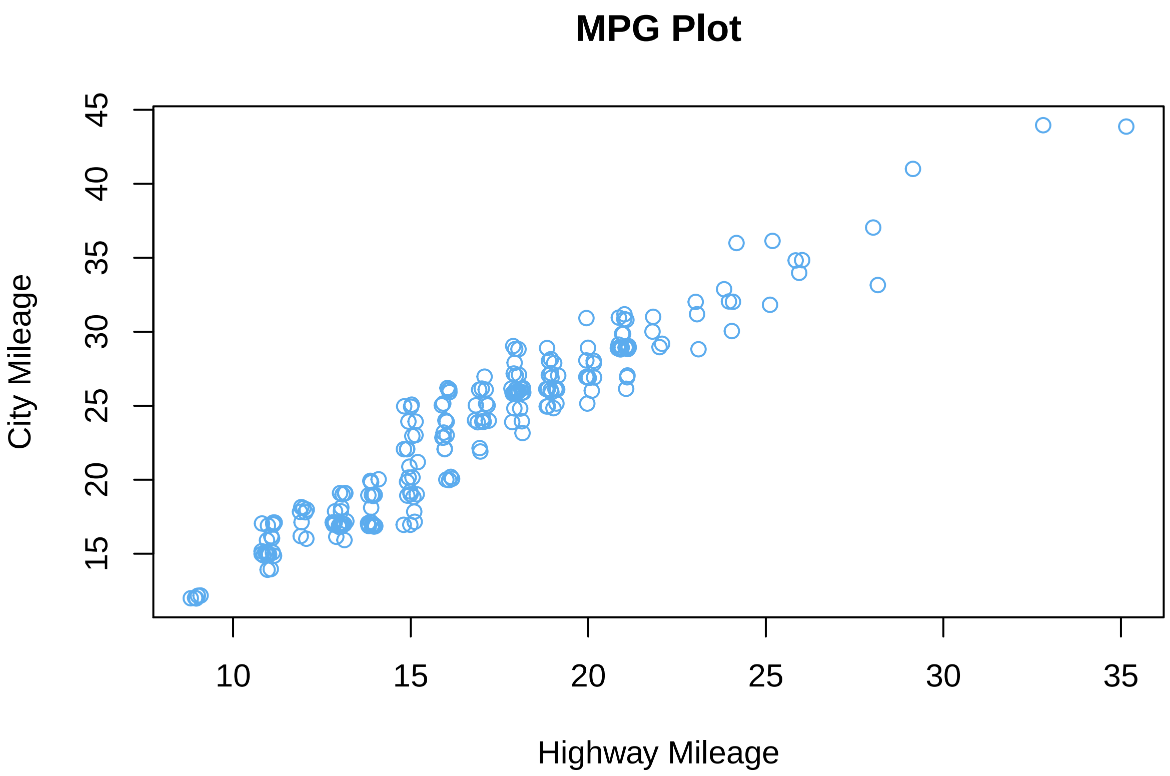
```
attach(mpg) #Attach the data frame so you can call columns by name
              # e.g., other wise you would need to type mpg$hwy instead of just hwy


plot(cty, hwy, xlab = 'Highway Mileage', ylab = 'City Mileage', main = 'MPG Plot')
```



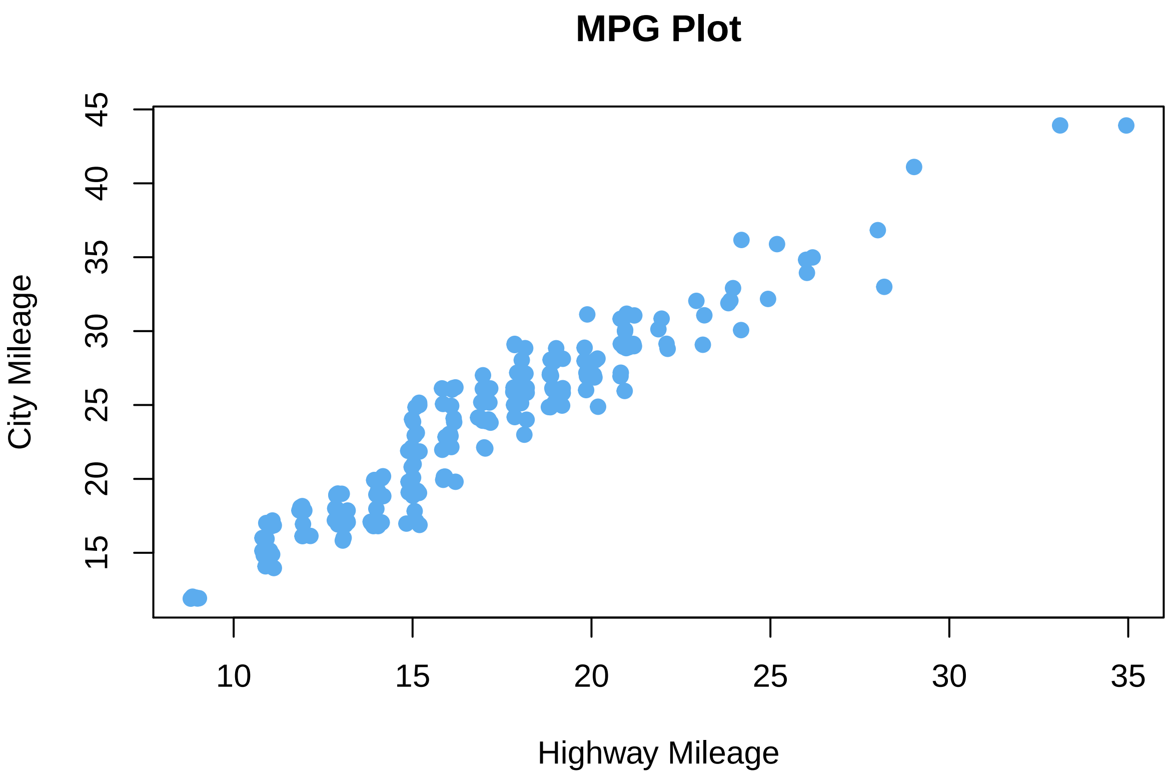
```r
# Jitter the points so they don't all overlap
plot(jitter(cty), jitter(hwy), xlab = 'Highway Mileage', ylab = 'City Mileage',
     main = 'MPG Plot')
```



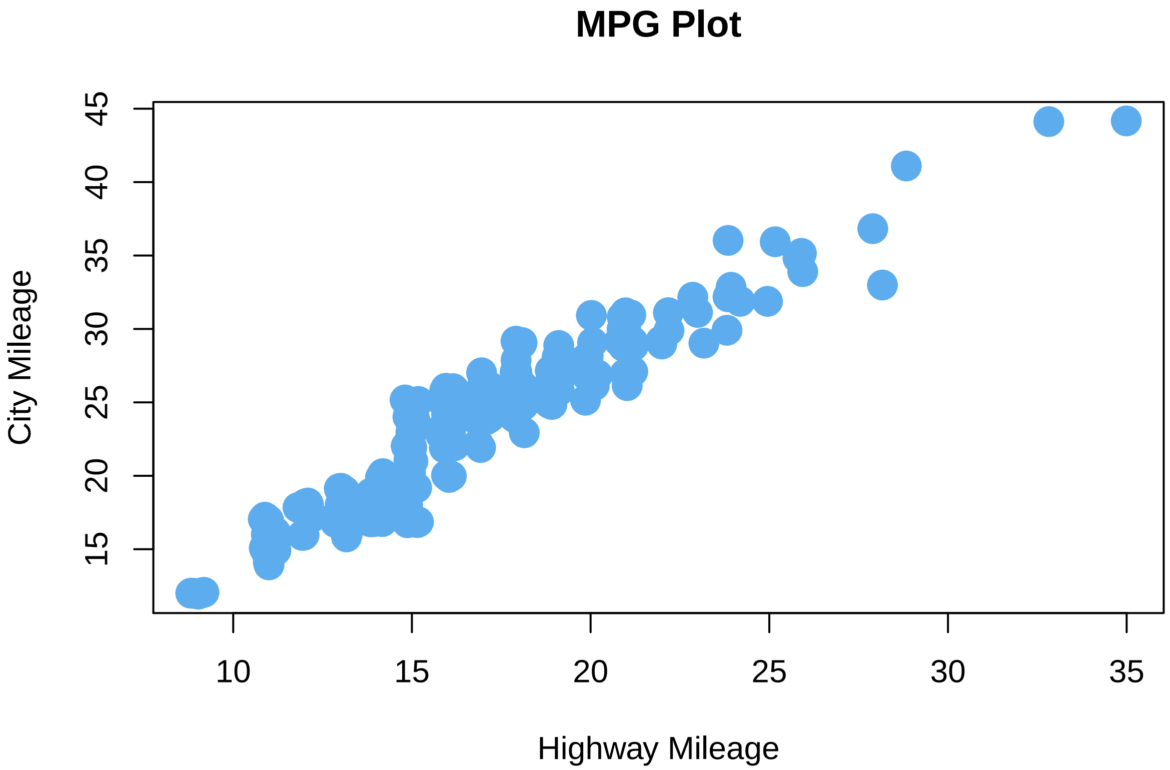
```
#Change the color
plot(jitter(cty), jitter(hwy), xlab = 'Highway Mileage', ylab = 'City Mileage',
     main = 'MPG Plot', col = 'steelblue2')
```



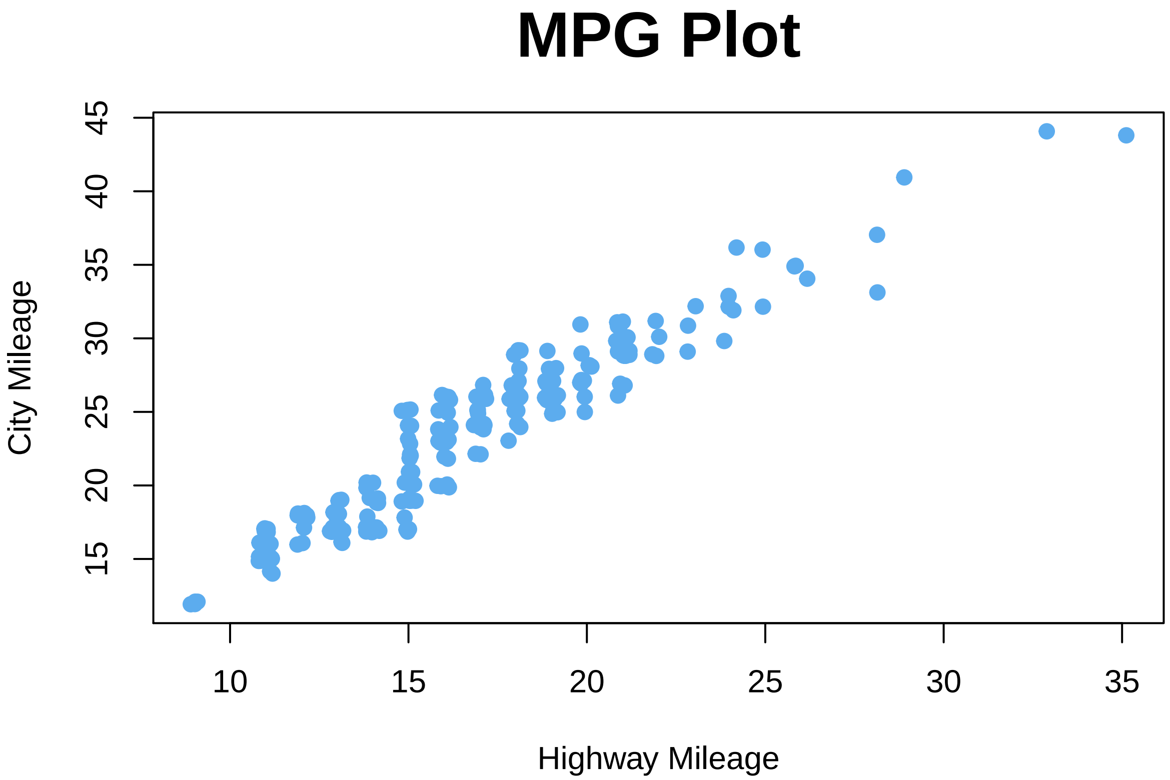
```
#Change the point type
plot(jitter(cty), jitter(hwy), xlab = 'Highway Mileage', ylab = 'City Mileage',
     main = 'MPG Plot', col = 'steelblue2', pch = 19)
```



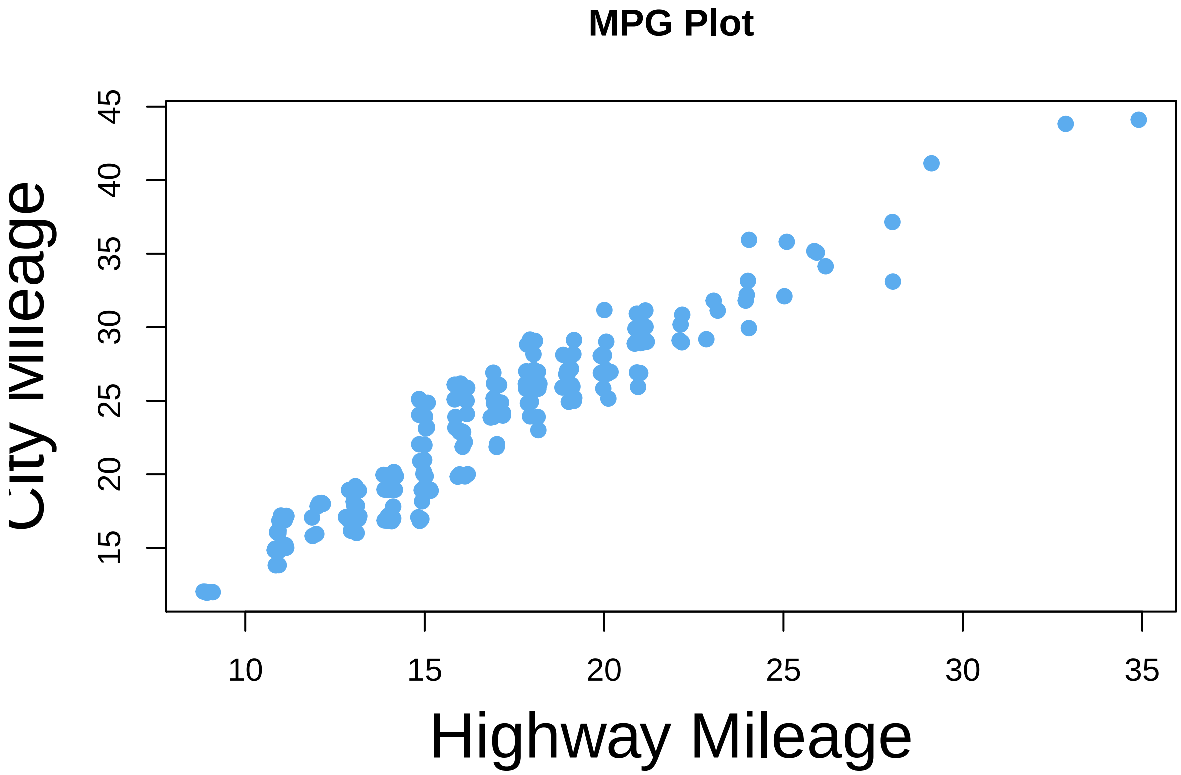
```
# See ?points for a list of point characters

#Change the point size
plot(jitter(cty), jitter(hwy), xlab = 'Highway Mileage', ylab = 'City Mileage',
     main = 'MPG Plot', col = 'steelblue2', pch = 19, cex = 2)
```



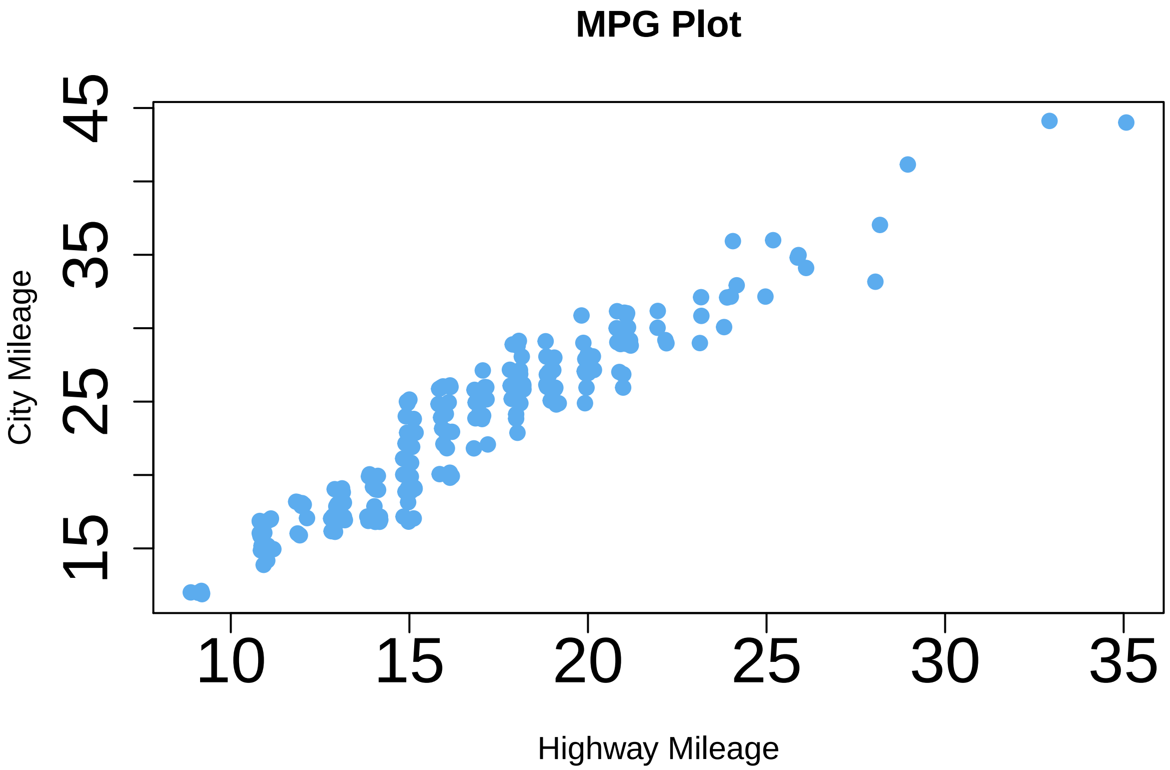
```r
# Change the text size of the title, axis labels (words) and axis ticks (numbers on tick marks)
#Change the point type
plot(jitter(cty), jitter(hwy), xlab = 'Highway Mileage', ylab = 'City Mileage',
     main = 'MPG Plot', col = 'steelblue2', pch = 19, cex.main = 2) #title size
```



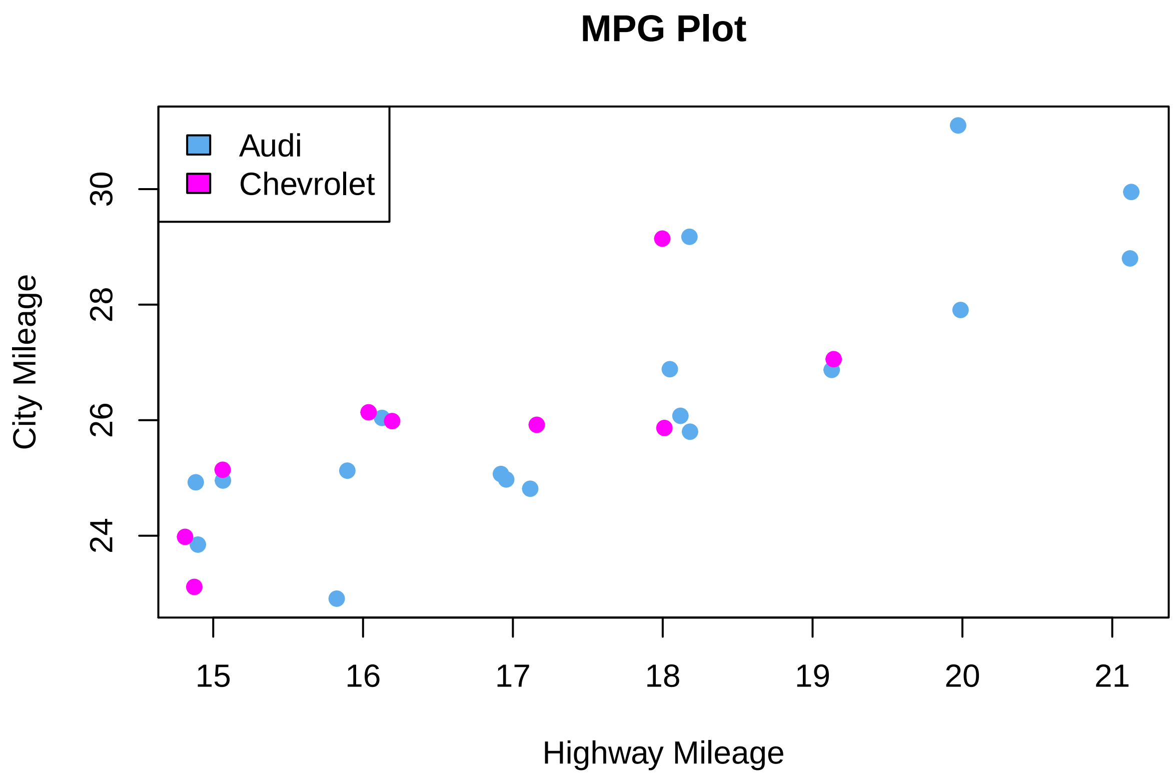
```r
plot(jitter(cty), jitter(hwy), xlab = 'Highway Mileage', ylab = 'City Mileage',
     main = 'MPG Plot', col = 'steelblue2', pch = 19, cex.lab = 2) #Axis label
```



```r
plot(jitter(cty), jitter(hwy), xlab = 'Highway Mileage', ylab = 'City Mileage',
     main = 'MPG Plot', col = 'steelblue2', pch = 19, cex.axis = 2) #Axis tick
```



```
# Keep your memory clear of clutter by detaching data frames when
#you are done plotting
detach(mpg)


# Plot Audi and Chevrolet data in different colors

#First, take subsets of the full data set
Audi = subset(mpg, subset = manufacturer == 'audi')
Chevrolet = subset(mpg, subset = manufacturer == 'chevrolet')

#Start by plotting the Audi data
plot(jitter(Audi$cty), jitter(Audi$hwy), xlab = 'Highway Mileage', ylab = 'City Mileage', main = 'MPG Pl

#Overlay the chevy plot using "points" or "lines"
points(jitter(Chevrolet$cty), jitter(Chevrolet$hwy), pch = 19, col = 'magenta')

#Add a legend in the top left
legend('topleft', legend =  c('Audi', 'Chevrolet'), fill = c('steelblue2', 'magenta'))
```



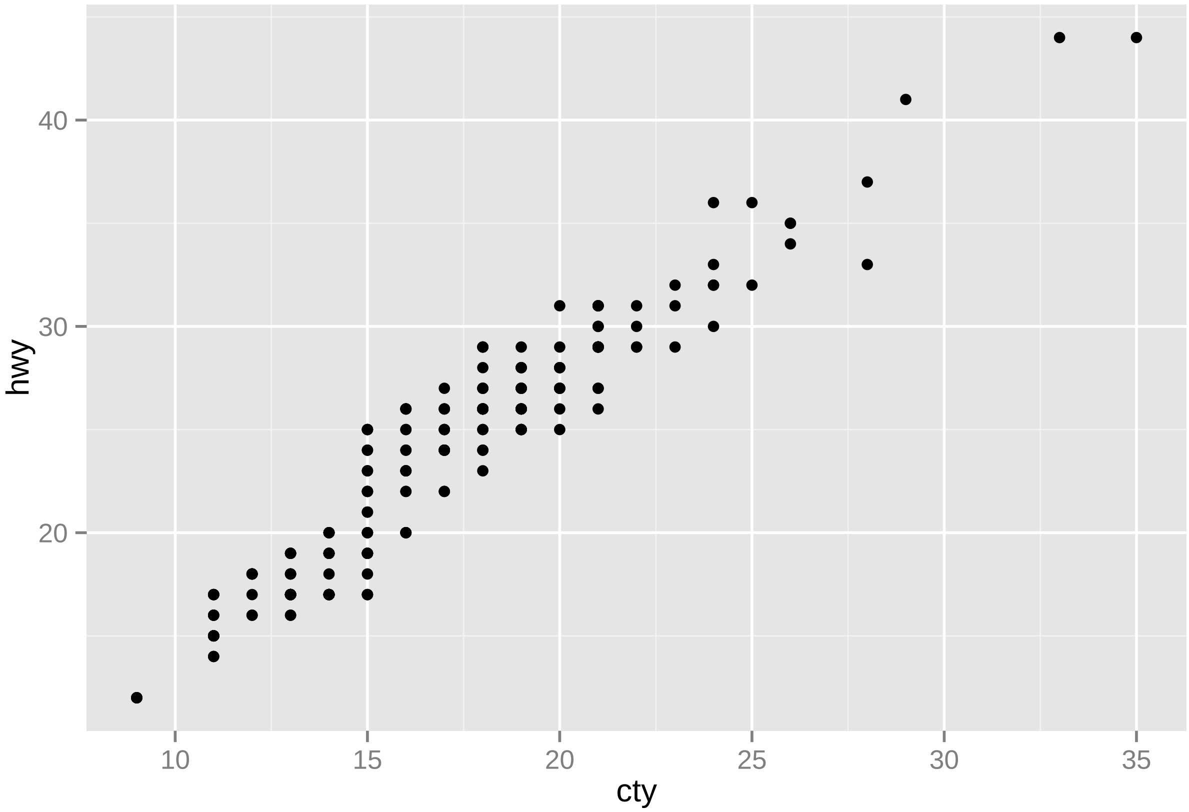
```
?legend
```

Section 2. The same examples, plotted using ggplot2

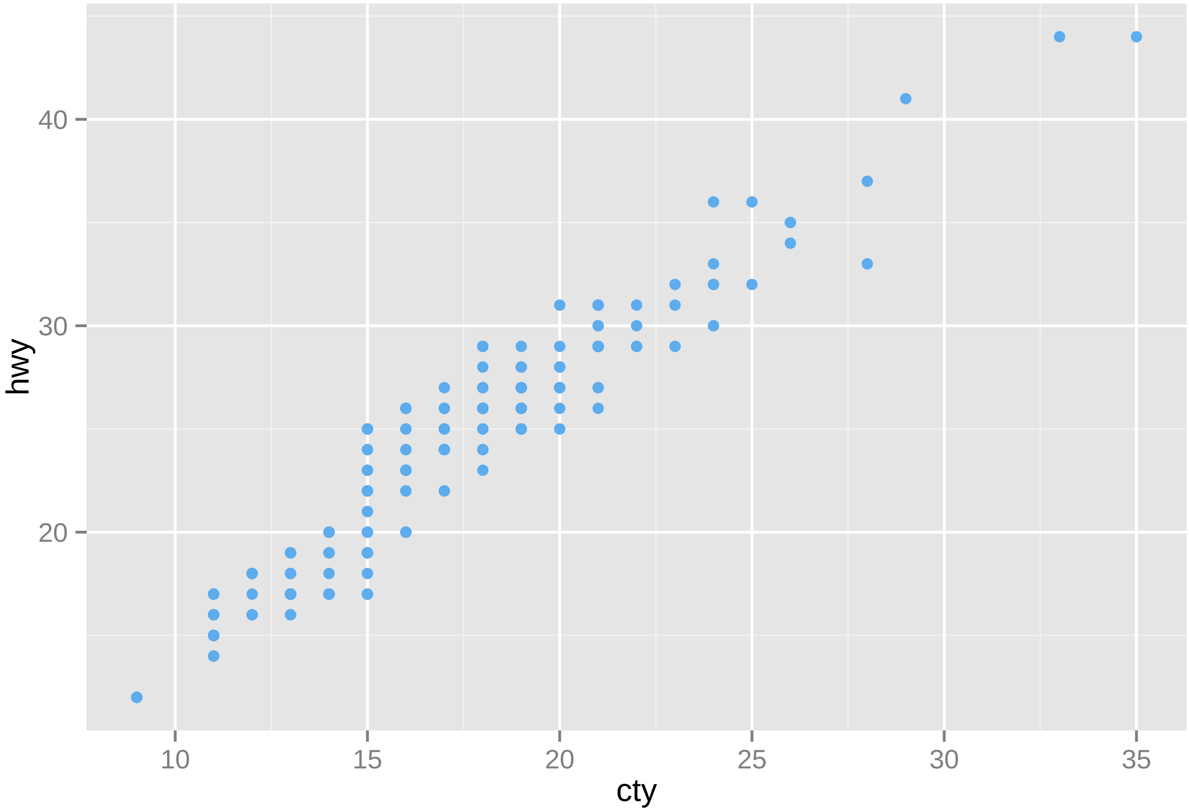
```
# Make a basic plot
p = ggplot(mpg, aes(x = cty, y = hwy))
```

NOTE: you MUST put your data into a data frame to use ggplot2 In ggplot syntax, you start with the name of your data frame, here mpg, and use the aesthetics option to tell it which columns to use on the x and y axis. Save these instructions to variable p. Then plot the basic plot layer, p below, using the + sign to add new plot layers (below layers include plot geometry, titles, etc.)

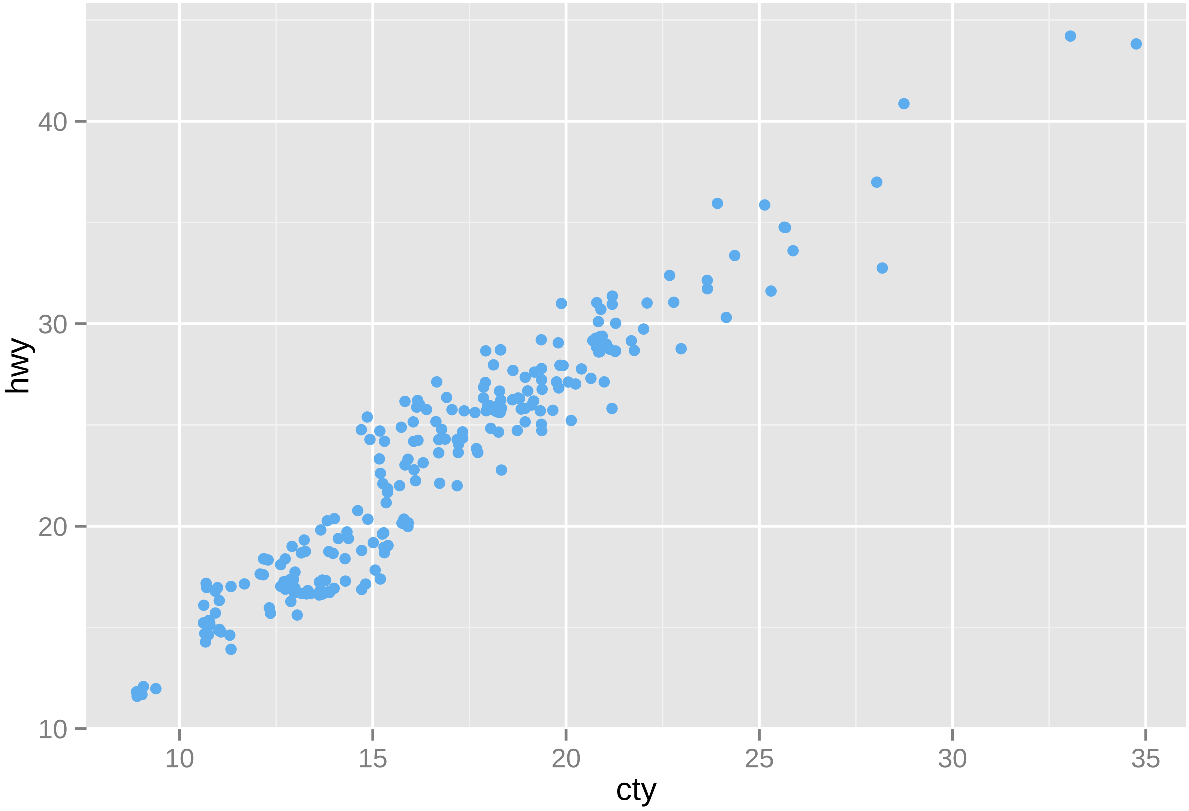
```
p + geom_point() #Show plot with basic geometry
```



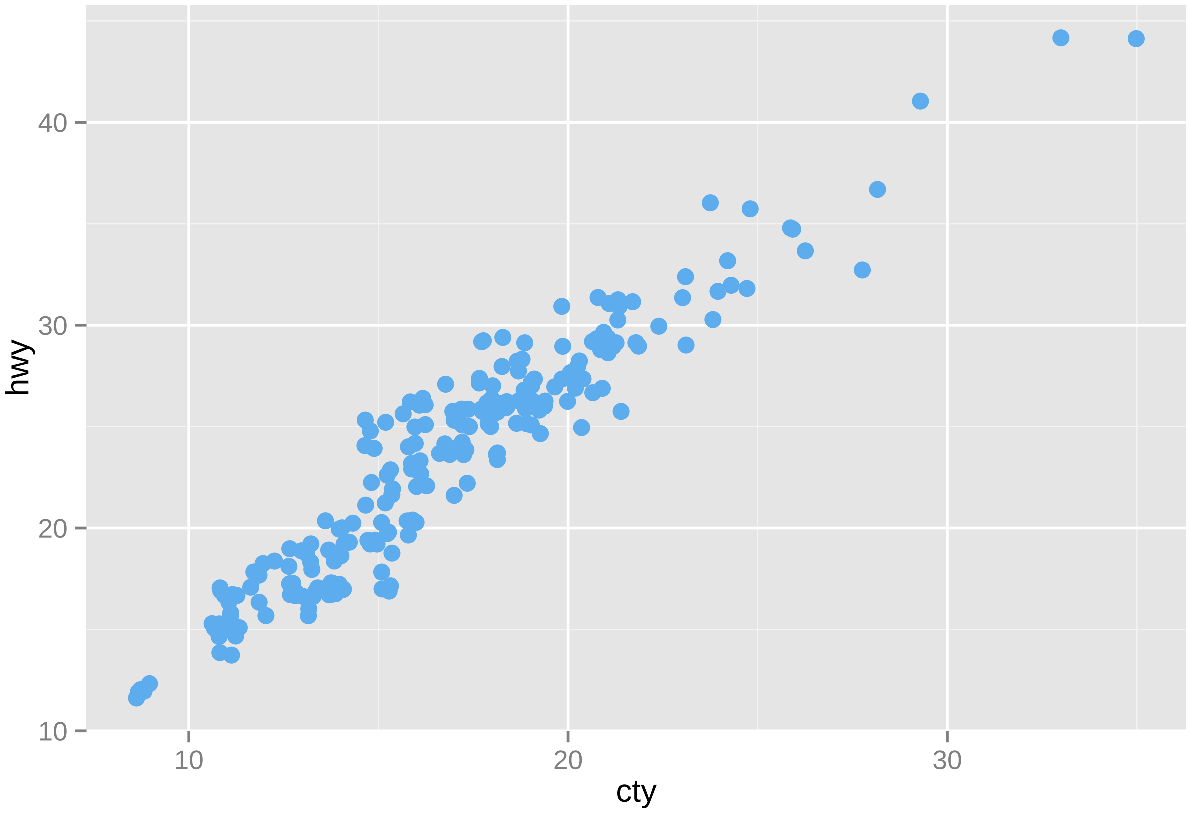
```
p + geom_point(color = 'steelblue2') #Add color
```



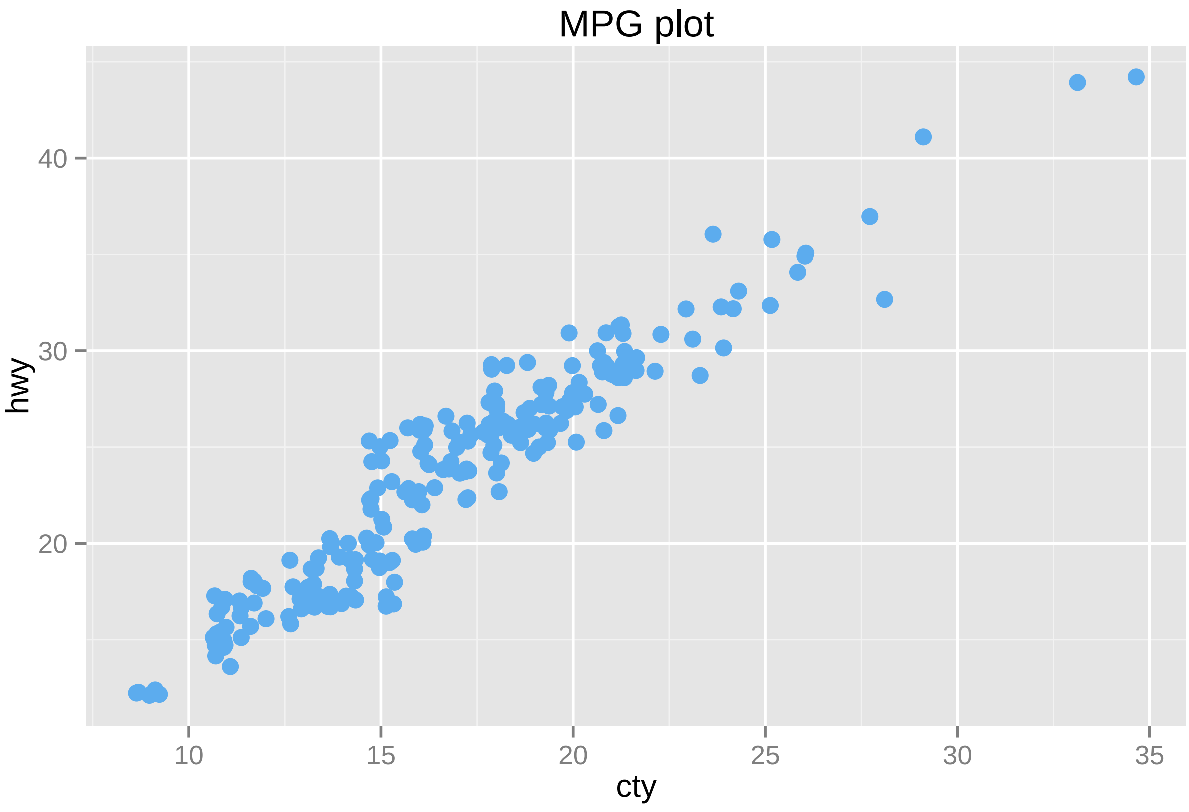
```
p + geom_jitter(color = 'steelblue2') #Add jitter
```



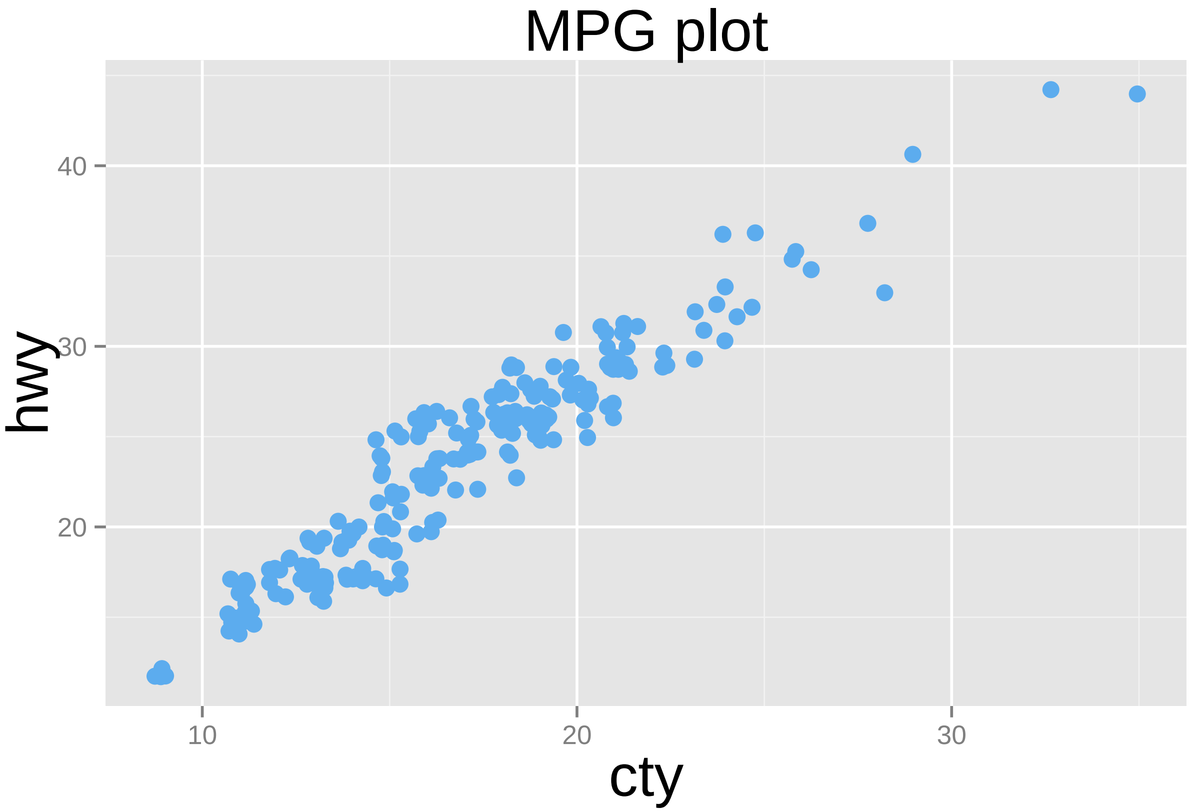
```
p + geom_jitter(color = 'steelblue2', size = 3) #Change point size
```



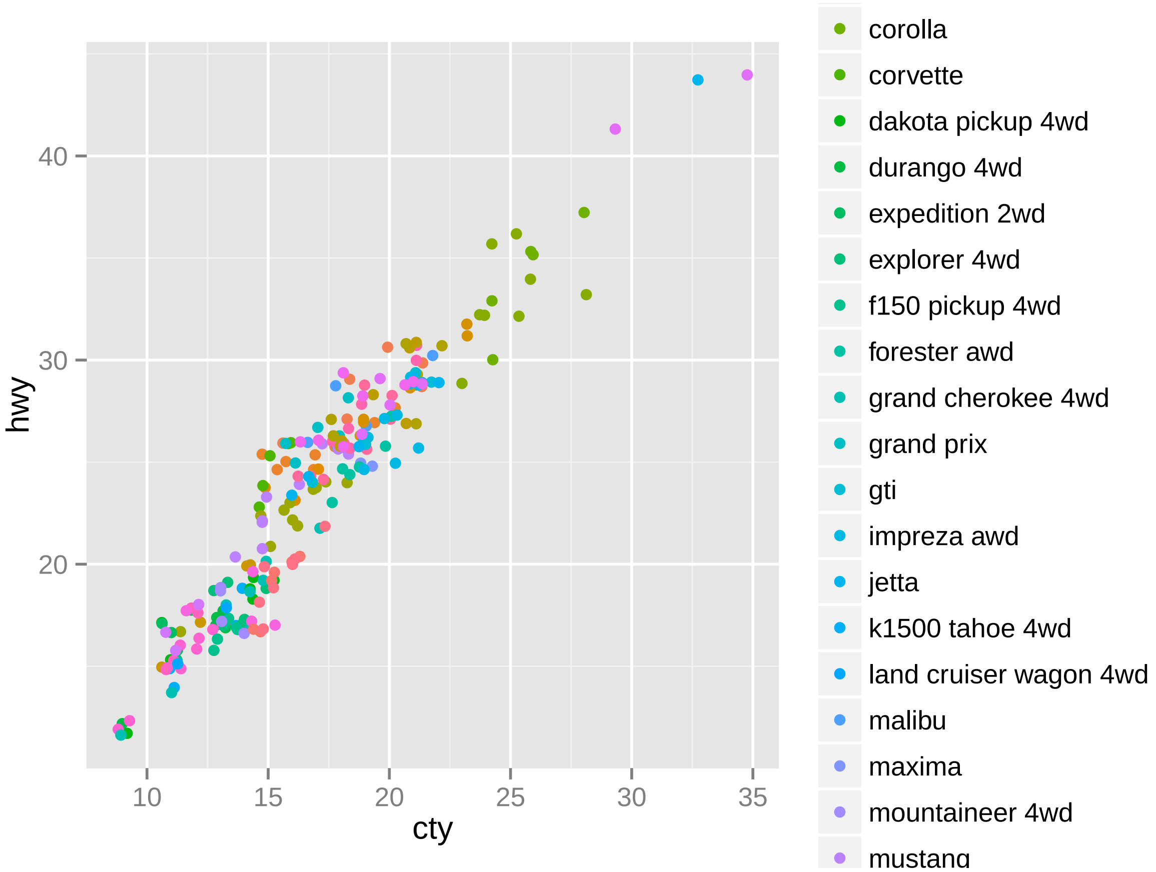
```
p + geom_jitter(color = 'steelblue2', size = 3) + ggtitle('MPG plot') #Add title
```



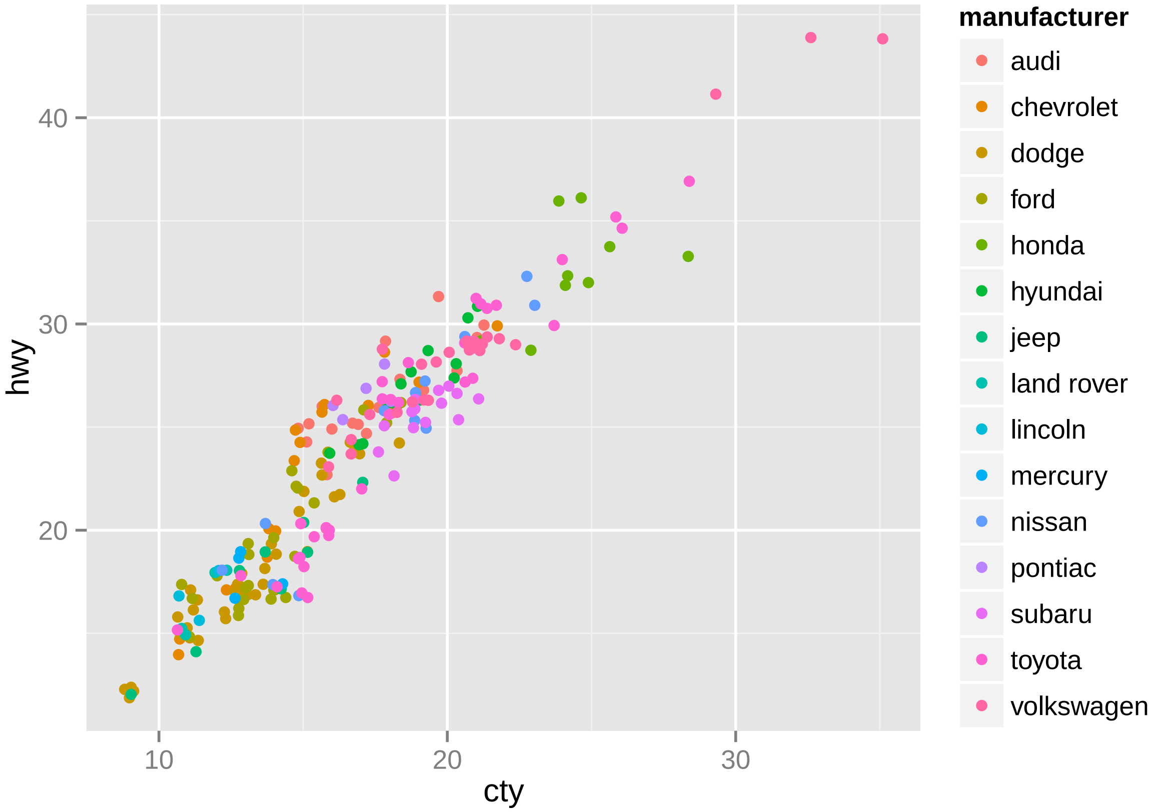
```
p + geom_jitter(color = 'steelblue2', size = 3) + ggtitle('MPG plot') +
  theme(axis.title = element_text(size=22)) +
  #Change axis label size
  theme(plot.title = element_text(size=22))
```



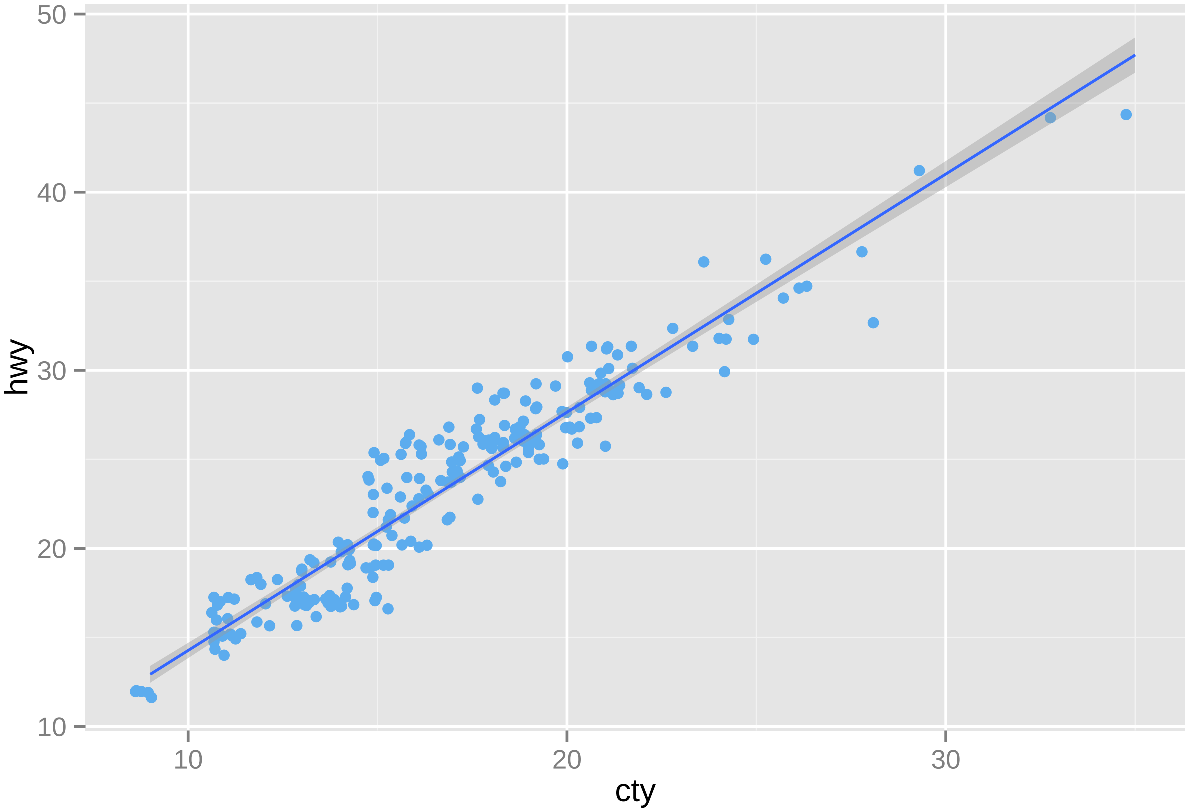
```
#Change plot title size

p + geom_jitter(aes(color = model)) # Determine point color based on
```



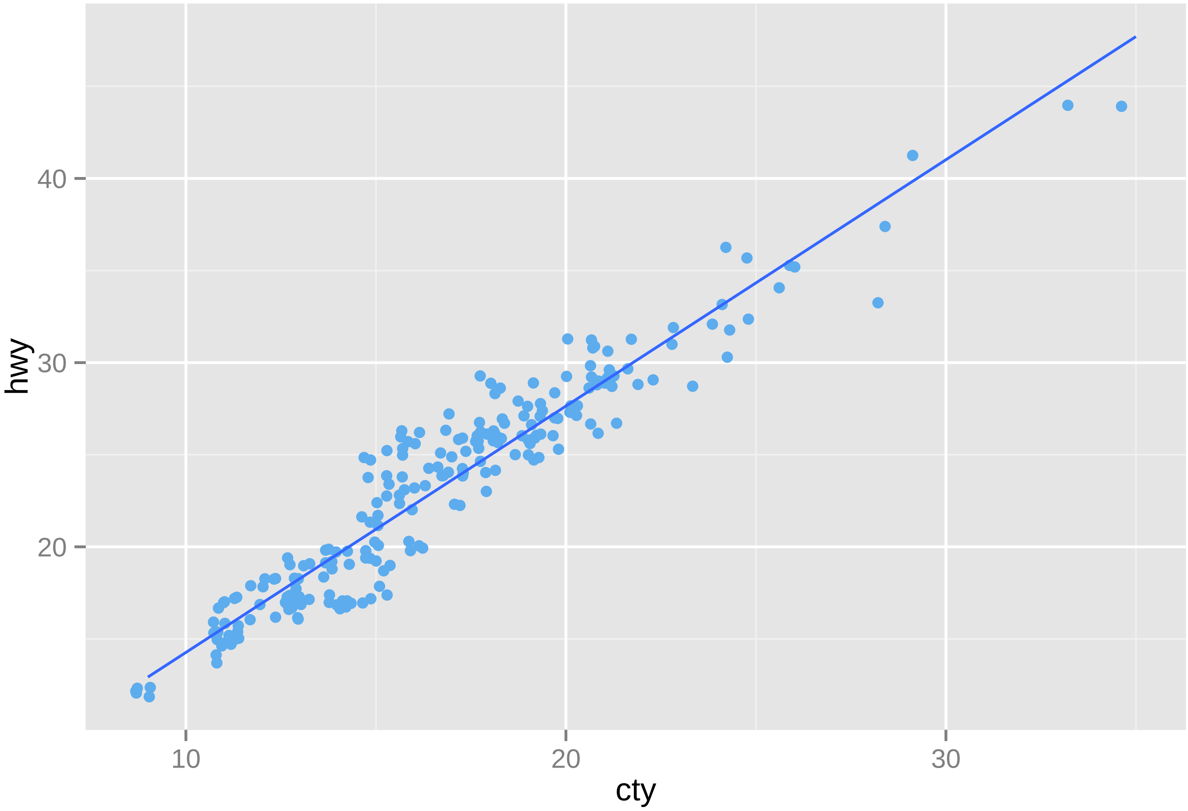
```
#the values in the "model" column

p + geom_jitter(aes(color = manufacturer)) # Determine point color based
```



```
#on the values in the "manufacturer" column

p + geom_jitter(color = 'steelblue2') + geom_smooth(method = lm) # Add a
```



```
#regression line with confidence intervals automatically plotted

p + geom_jitter(color = 'steelblue2') + geom_smooth(method = lm, se = FALSE)
```



```
# Add a regression line without confidence intervals
# Note, you use similar modifications to change the point size

## See the ggplot documentation for help, or Google for example
#scripts that make the plots you want
```

This page was made using R Markdown with package knitr.