

## Brief introduction to the ggplot2 package via the qplot() function

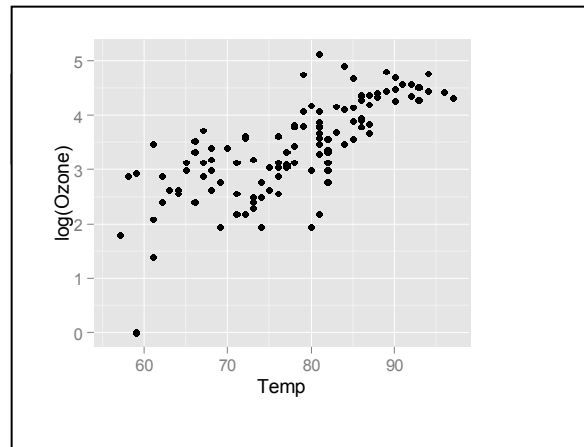
Hadley Wickham, the creator of **ggplot2**, advises that one of the best ways to learn how to use **ggplot2** is to start small by relying on the function **qplot()**. Here, **qplot** is a shortcut for quick plot.

For instance, to create a basic scatterplot of the variables **Ozone** (log-transformed) and **Temp** in the data frame **airquality** available in R, we would use these commands:

```
install.packages("ggplot2")
```

```
require(ggplot2)
```

```
qplot(Temp, log(Ozone), data=airquality)
```



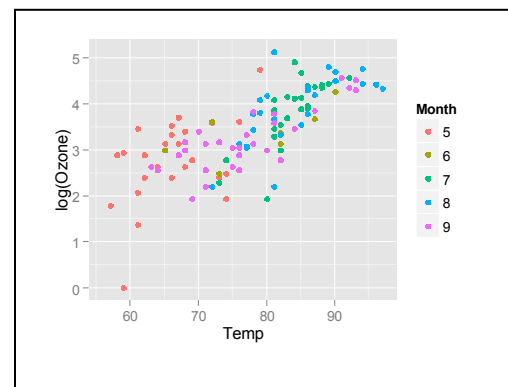
Any basic scatterplot produced by the **qplot()** function can be enhanced in various ways. For instance, we could control the *shape*, *colour* or *size* of the data points displayed in the scatterplot using so-called *aesthetics* (also known as *aes*). We could also use *faceting* to display different subsets of the data.

As an example, let us convert the **Month** variable in the **airquality** data frame into a qualitative variable using the **factor()** function. Once this is accomplished, we can display the relationship between **log(Ozone)** and **Temp** separately for each month represented in the **Month** variable.

```
airquality$Month <- factor(airquality$Month)
```

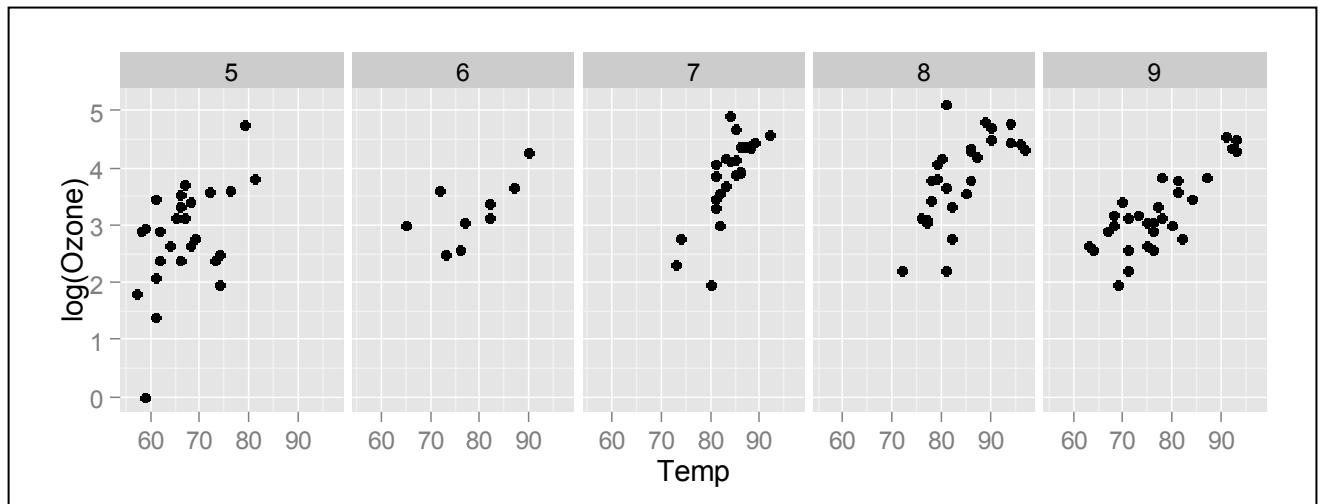
```
str(airquality)
```

```
qplot(Temp, log(Ozone), colour=Month, data=airquality)
```



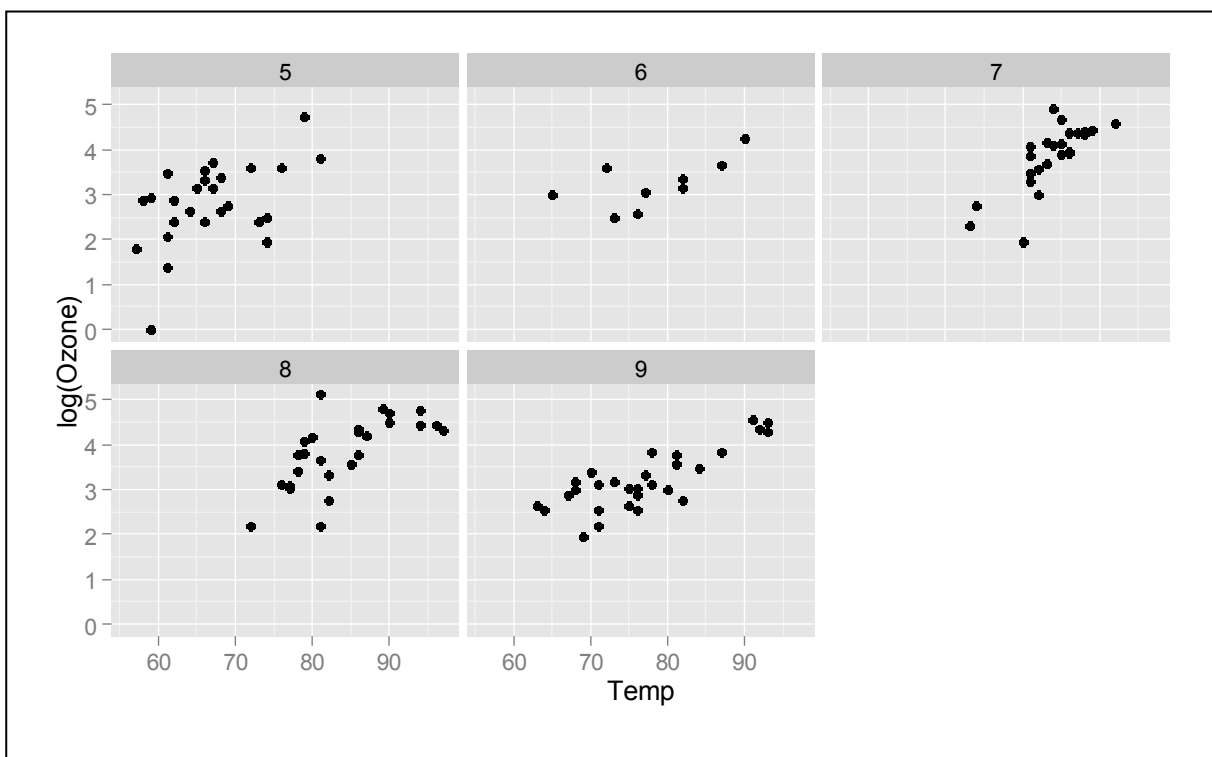
If we find that the previous scatterplot is too cluttered to read properly, we can display relationship between **log(Ozone)** and **Temp** conditional on **Month** in separate panels. This can be done by ‘adding’ the function `facet_grid()` to `qplot()`. Note the use of the syntax `.~ Month` inside the `facet_grid()` function, which simply means “separately for each Month”.

```
qplot(Temp, log(Ozone), data=airquality) + facet_grid(. ~ Month)
```



A slight variation of the previous R command is given below. This variation uses `facet_wrap()` instead of `facet_grid()` to force R to “wrap” the scatterplot panels in the graphical window and thus use the graphical space more efficiently. Note that `facet_wrap()` uses the syntax `~ Month` instead of `.~Month`.

```
qplot(Temp, log(Ozone), data=airquality) + facet_wrap(~ Month)
```

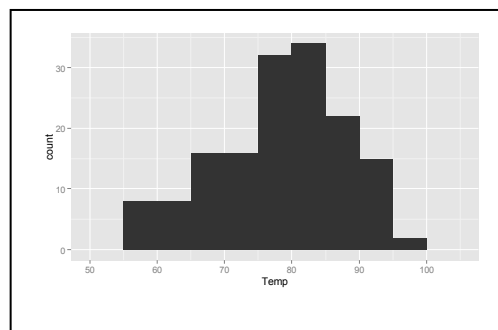


By default, if `qplot()` is used with two variables (e.g., `log(Ozone)` and `Temp`), it will produce a scatterplot. However, if `qplot()` is used with a single variable (e.g., `Temp`), it will produce a histogram. This is illustrated by the command below:

```
qplot(Temp, data=airquality)
```

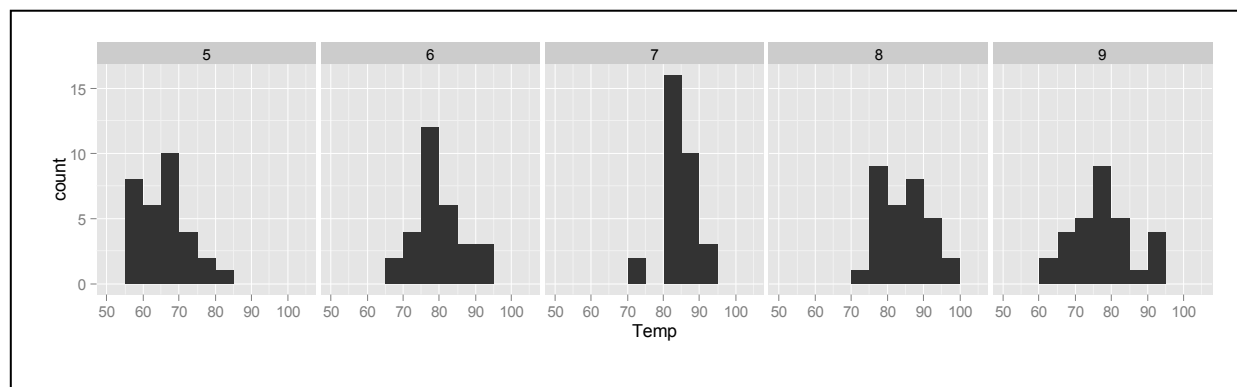
By default, histograms displayed by `qplot` use a binwidth equal to the range of the data used to construct the histogram divided by 30. Other user-specified binwidths can be specified as follows:

```
qplot(Temp, binwidth=5, data=airquality)
```



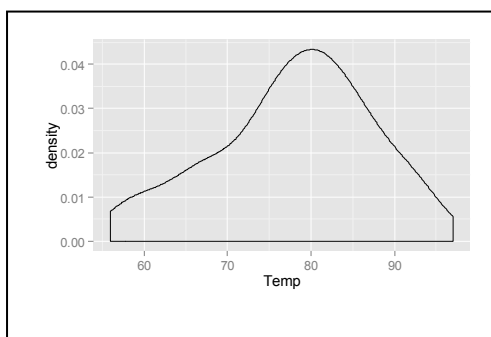
Once again, we can use `facet_grid()` in conjunction with `qplot()` to construct histograms of `Temp` separately for each `Month`.

```
qplot(Temp, binwidth=5, data=airquality) + facet_grid(. ~ Month)
```



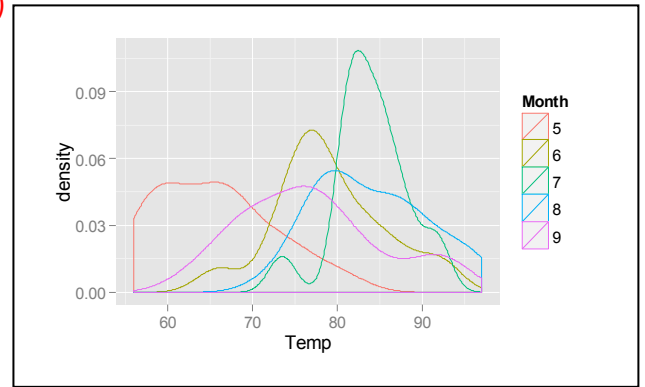
A density plot of the `Temp` variable can be obtained by using `qplot()` with the option `geom="density"`:

```
qplot(Temp, geom= "density", data=airquality)
```

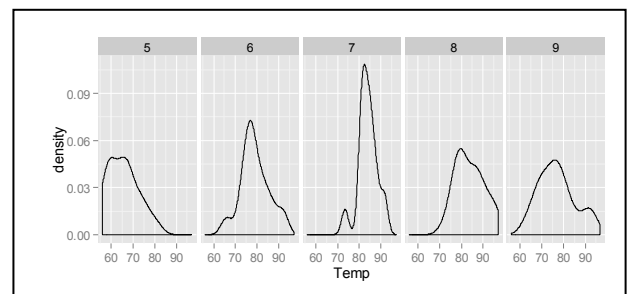


If we need to visualize the density of **Temp** separately for each **Month**, we have two options:

```
qplot(Temp, geom= "density", colour=Month, data=airquality)
```

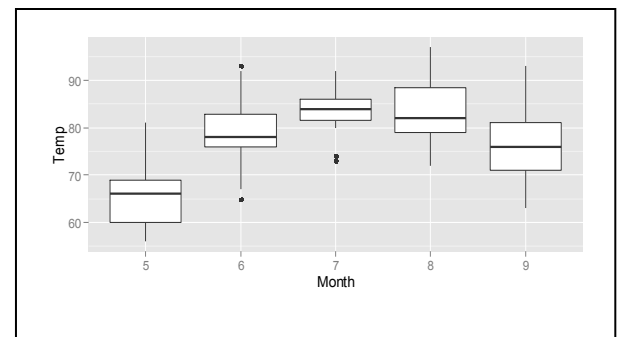


```
qplot(Temp, geom= "density", data=airquality) + facet_grid(. ~ Month)
```



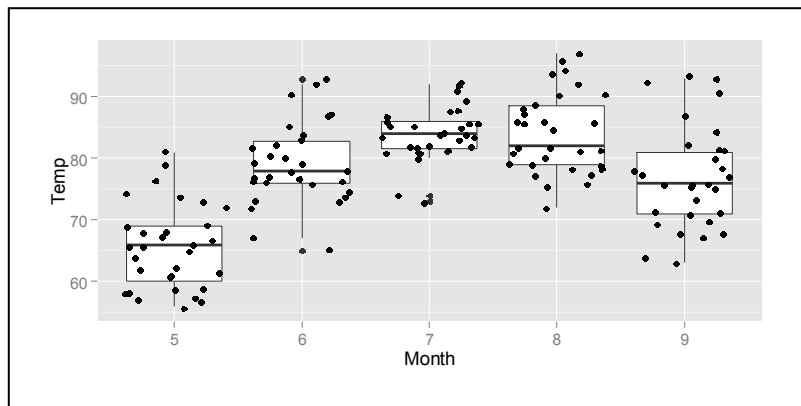
To produce side-by-side boxplots of **Temp** versus **Month**, use `qplot()` in conjunction with the option `geom="boxplot"`, as seen below:

```
qplot(Month, Temp, geom= "boxplot", data=airquality)
```



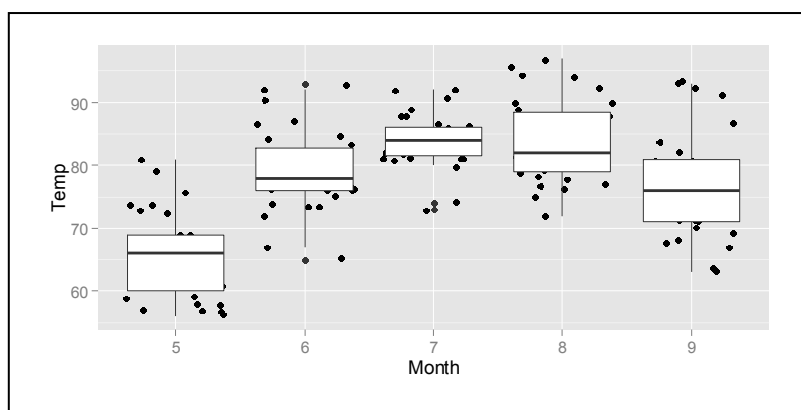
In side-by-side boxplots, it is possible to also display the actual data in each group (jittered):

```
qplot(Month, Temp, geom= c("boxplot","jitter"), data=airquality)
```



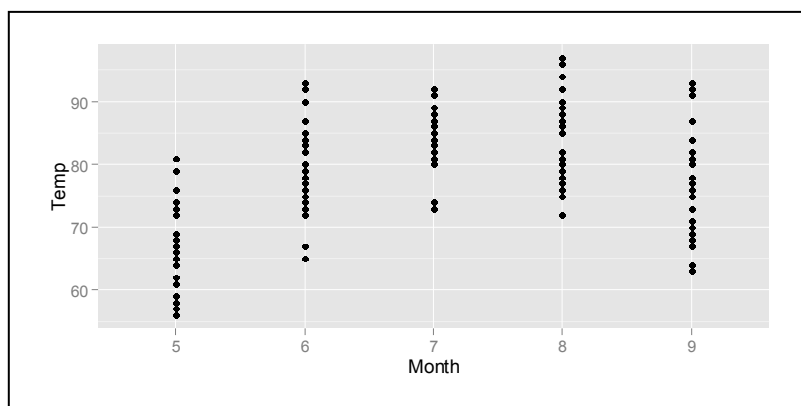
Switching the order in which the geom options "boxplot" and "jitter" are listed will place the actual data in each group behind the boxplot corresponding to that group:

```
qplot(Month, Temp, geom= c("jitter","boxplot"), data=airquality)
```



For the previous command, leaving out the geom options altogether will produce a slightly different display of the data in each group (i.e., a side-by-side stripchart):

```
qplot(Month, Temp, data=airquality)
```



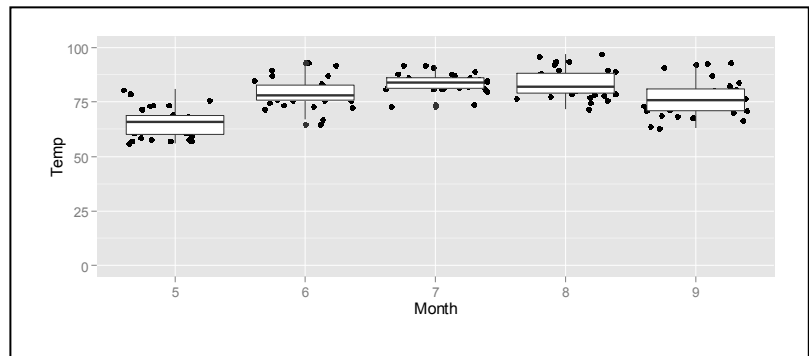
Of course, the data in the side-by-side stripchart can be jittered if necessary:

```
qplot(Month, Temp, geom= "jitter", data=airquality)
```

For any plot produced by `qplot()`, it is easy to change the x axis and y axis limits. Here is an example showing how to change the limits of the y axis for a side-by-side boxplot:

```
qplot(Month, Temp, geom= c("jitter", "boxplot"), data=airquality)
```

```
last_plot() + ylim(0,100)
```



Using `qplot()` it is also possible to construct a time series plot. For instance, we can plot the **Temp** values over time provided we create a **Date** variable to represent time and append it to the **airquality** data set:

```
airquality$Date <- ISOdate(1973, airquality$Month, airquality$Day)
```

```
qplot(Date, Temp, geom= c("line"), data=airquality)
```

In the above, we used the fact that the **airquality** data were collected in 1973 and that we knew the month and day corresponding to each observation in the data set.

