Hands-On Data Science with R
Miscellaneous Plots in R

Graham.Williams@togaware.com

26th August 2014

Visit http://HandsOnDataScience.com/ for more Chapters.

In this chapter we explore a variety of plots generated using R. For an introduction to plots based on ggplot2 see the specific GGPlot2 chapter.

The required packages for this module include:

```
library(iplots)
library(ggplot2)
library(tabplot)
library(rattle)
library(dplyr)
```

As we work through this chapter, new R commands will be introduced. Be sure to review the command's documentation and understand what the command does. You can ask for help using the `?` command as in:

```
?read.csv
```

We can obtain documentation on a particular package using the *help=* option of `library()`:

```
library(help=rattle)
```

This chapter is intended to be hands on. To learn effectively, you are encouraged to have R running (e.g., RStudio) and to run all the commands as they appear here. Check that you get the same output, and you understand the output. Try some variations. Explore.

1 Dial Plot

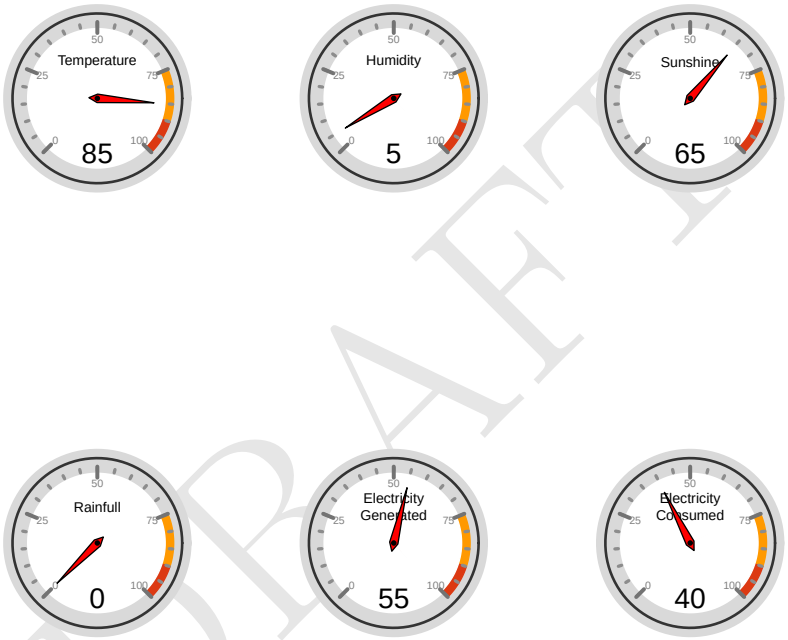
The dial plot is available as the Google Gauge Plot. Pentaho Business Intelligence provides the dial plot for dashboards However, Hadley Wickham suggests we "are trying to understand your data, not driving a racing car or aeroplane." Hadley points us to the work of Stephen Few who presents the "powerful and eloquent" arguments and suggests alternatives for the most effective presentation of data.

Nonetheless, Gaston Sanchez wrote and Jeff Hemsley modified a version of `dial.plot()` for R.

```
source("http://onepager.togaware.com/dial.plot.R")
dial.plot(label="OnePageR", value=85)
```

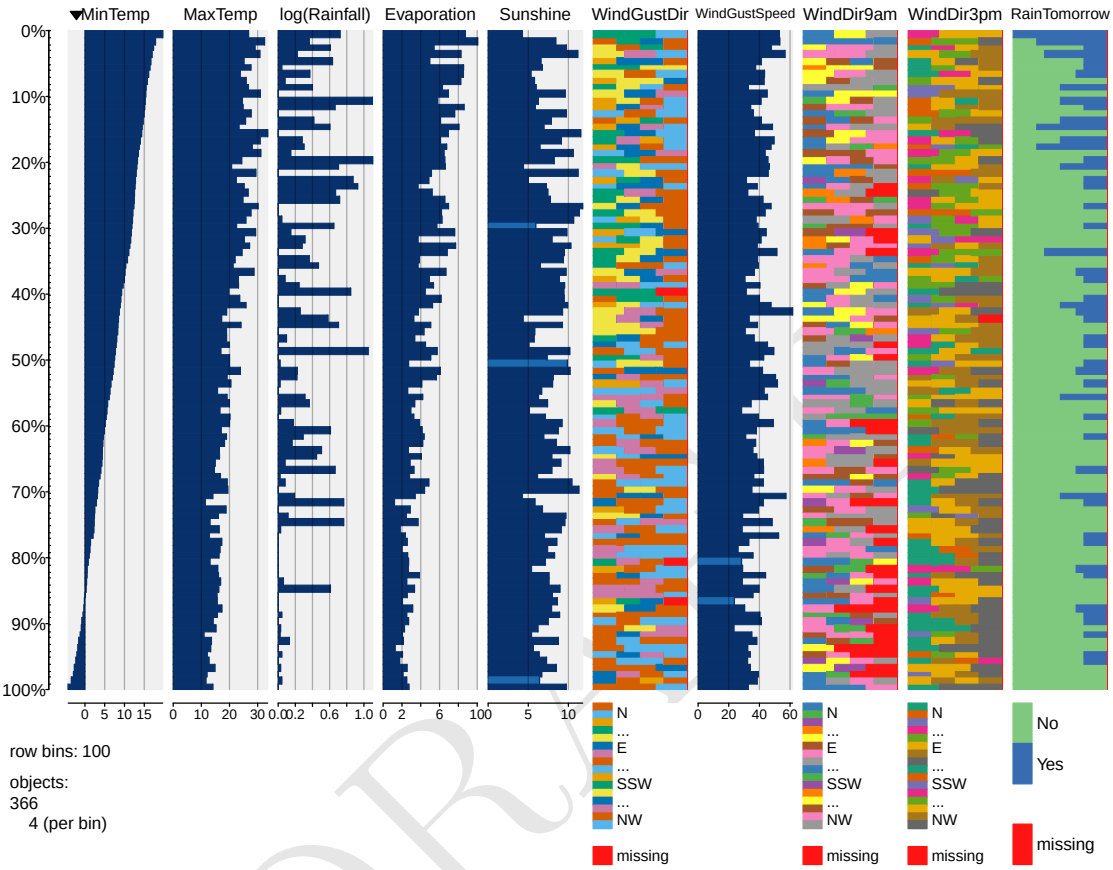
2 Dashboard

```
opar <- par(mfrow=c(2,3))
dial.plot(label="Temperature", label.cex=1, value=85, value.cex=2)
dial.plot(label="Humidity", label.cex=1, value=5, value.cex=2)
dial.plot(label="Sunshine", label.cex=1, value=65, value.cex=2)
dial.plot(label="Rainfull", label.cex=1, value=0, value.cex=2)
dial.plot(label="Electricity\nGenerated", label.cex=1, value=55, value.cex=2)
dial.plot(label="Electricity\nConsumed", label.cex=1, value=40, value.cex=2)
```



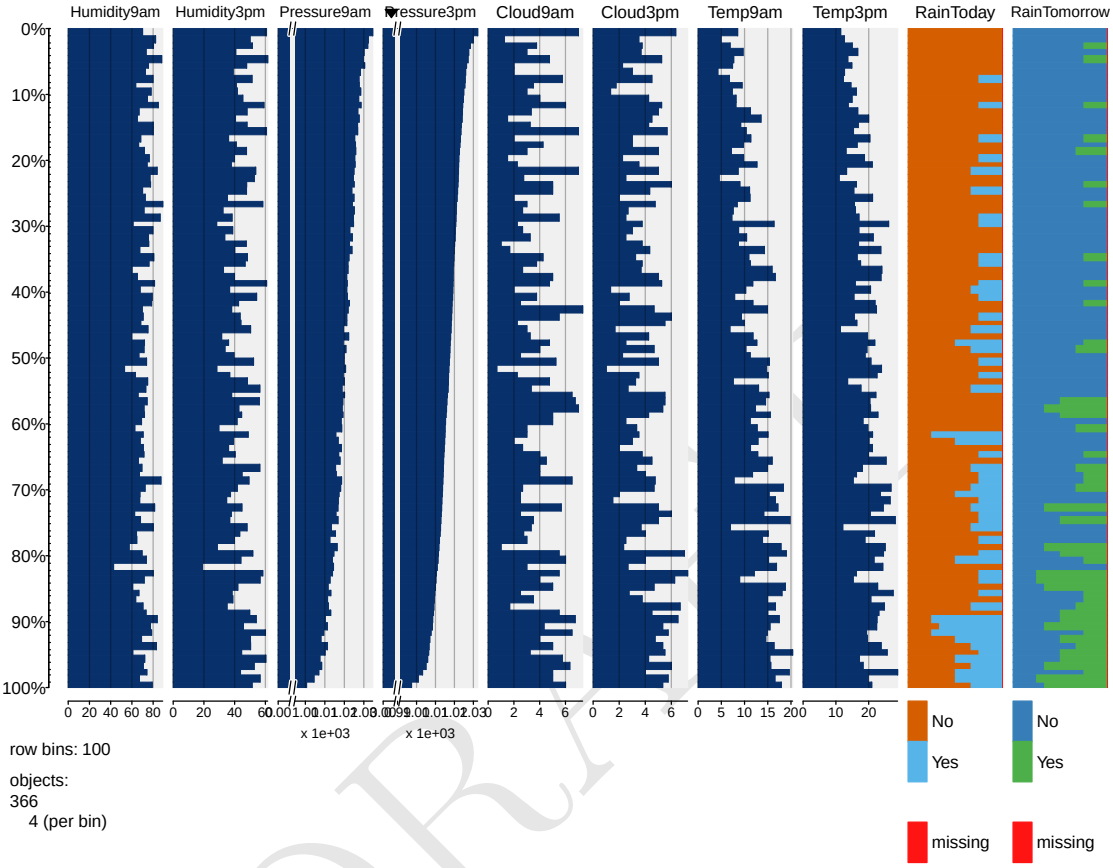
3 Table Plots

```
library(tabplot)
tableplot(weather, select=c(MinTemp:WindDir3pm, RainTomorrow))
```



4 Table Plots

```
library(tabplot)
tableplot(weather, select=c(Humidity9am:RainToday, RainTomorrow), sortCol=Pressure3pm)
```



5 Visually Weighted Regression

From Nicrebread www.nicebread.de (and posted on Bloggers on R) by Felix Schoenbrodt 30 August 2012 addressing Solomon Hsiang's proposal of an appealing method for visually displaying the uncertainty in regressions and using shading in response to Gelman's note that traditional statistical summaries such as 95% intervals give too much weight to the edges.

6 F1: Exploring the Dataset

We can now explore a particular dataset using ggplot2 graphics to get an understanding of the story behind the data. The data and the original plots (some are now modified) are from Tony Hirst's blog.

```
(load("data/f1.RData"))
```

```
## [1] "f1"
```

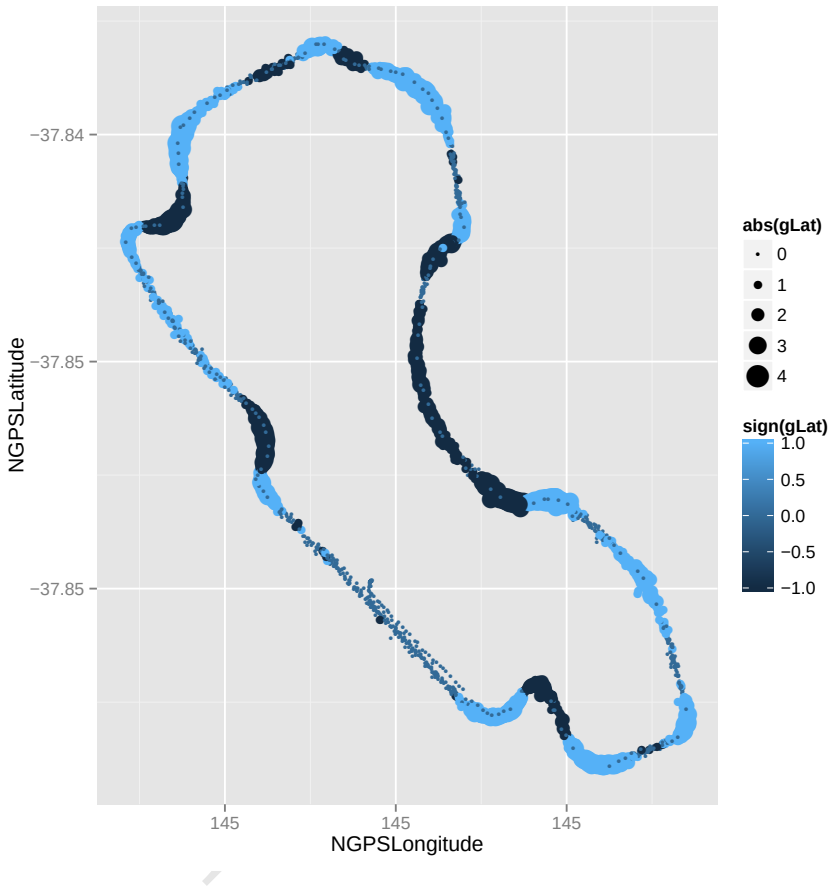
```
head(f1)
```

```
##         file timestamp NGPSLatitude NGPSLongitude NGear nEngine
## 1 1269758114  17:35:10       -37.85           145     4   13422
## 2 1269758115  17:35:11       -37.85           145     3   13383
## 3 1269758116  17:35:12       -37.85           145     3   14145
....
```

7 F1: Simple Map

We can draw the particular F1 circuit using the longitude and latitude as the x and y coordinates, and using the sign of the latitudinal g-force on the driver. We believe that a positive value of `gLat` indicates force to the left and a negative value indicates a force to the right.

```
p <- ggplot(f1, aes(NGPSLongitude, NGPSLatitude))
p <- p +  geom_point(aes(col=sign(gLat), size=abs(gLat)))
p
```



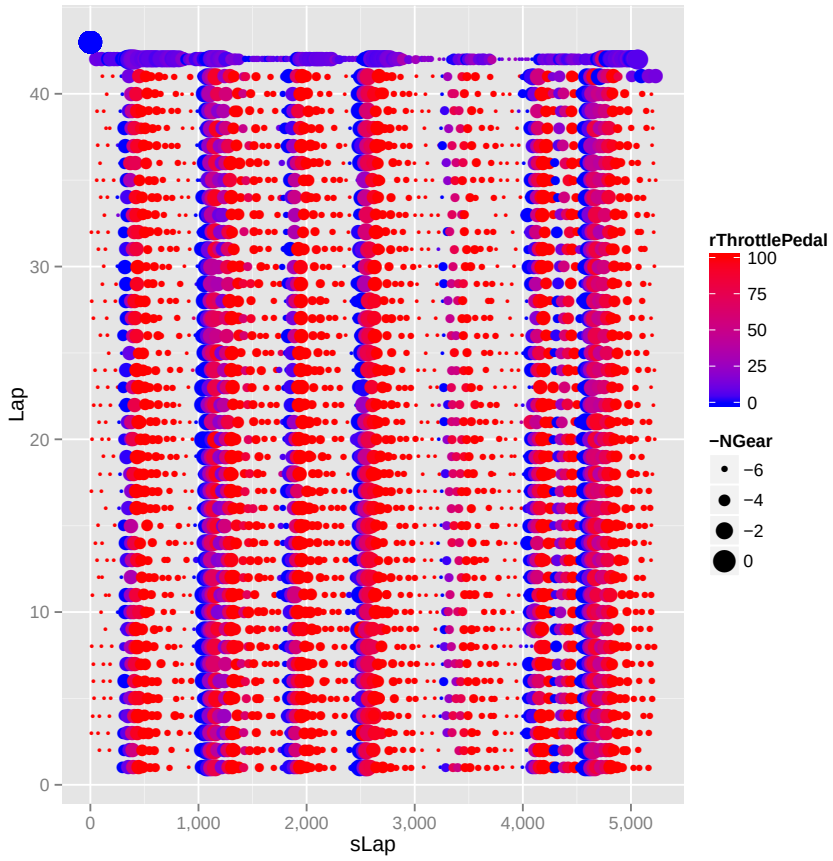
Based on Tony Hirst's Blog Post, March 2012

We should be able to see from the plot the forces on the driver on the left and right hand corners, and see how tight the corner is based on the size of the dots.

8 F1: Driver Behaviour

We can explore the driver's behaviour in using low gear and throttle. The distance around the track is plotted on the x-axis and the lap number on y axis. The node size is inversely proportional to gear number (low gear, large point size) and the colour is the relative amount of throttle pedal depression.

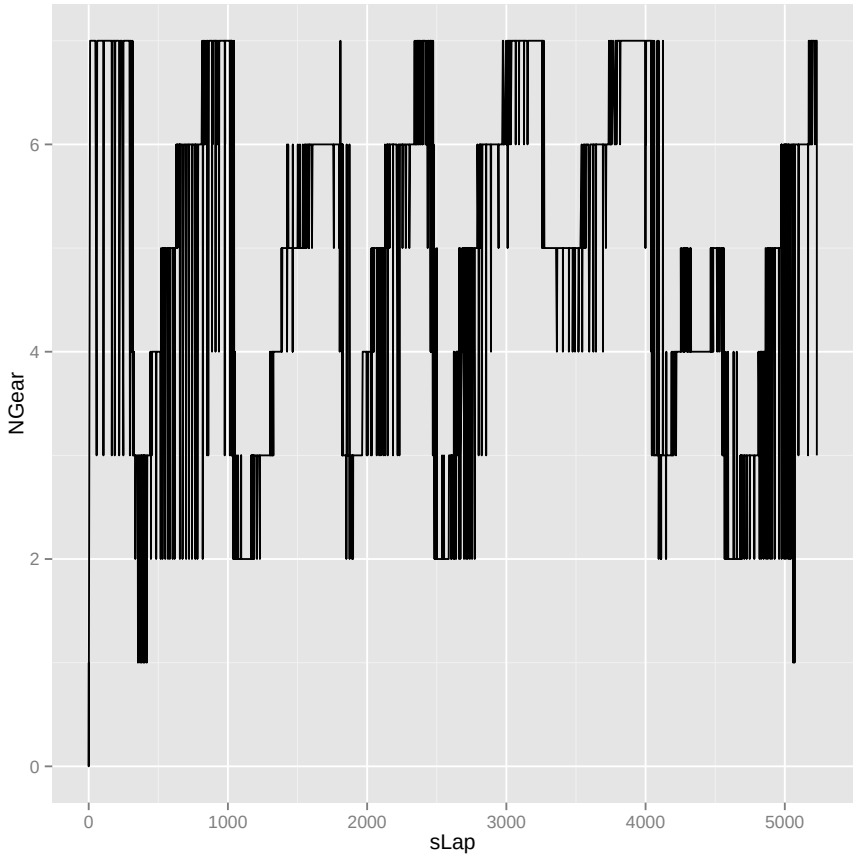
```
library(scales)
p <- ggplot(f1, aes(sLap, Lap))
p <- p + geom_point(aes(col=rThrottlePedal, size=-NGear))
p <- p + scale_colour_gradient(low="blue", high="red")
p <- p + scale_x_continuous(labels=comma)
p
```



Based on Tony Hirst's Blog Post, March 2012

9 F1: Gear Usage Around the Track

```
p <- ggplot(f1, aes(sLap, NGear))
p <- p + geom_line()
p
```

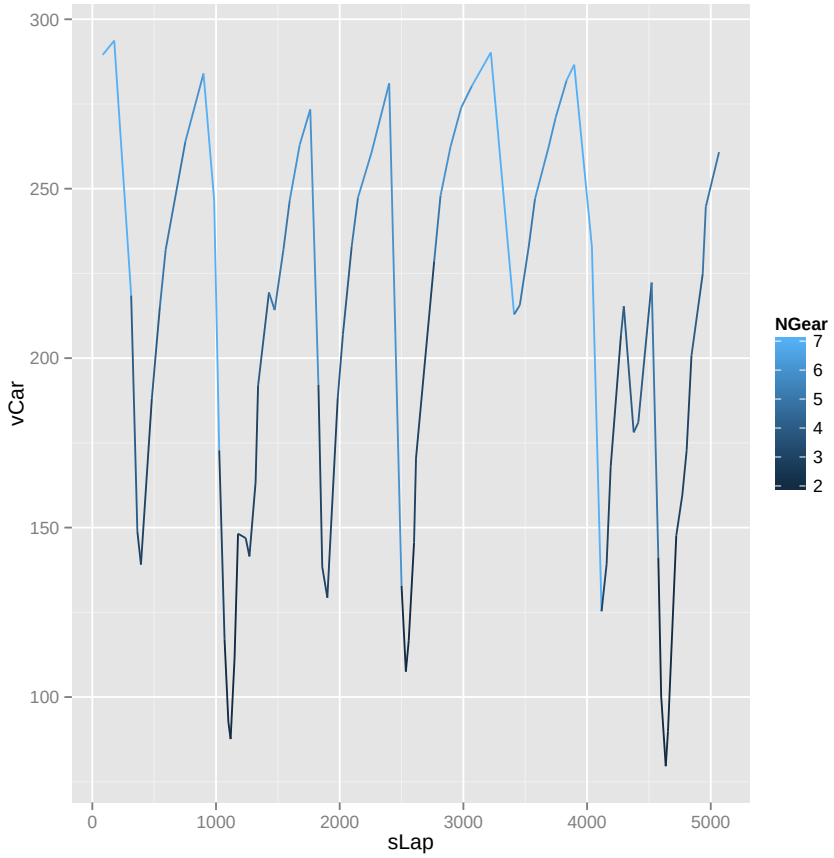


Based on Tony Hirst's Blog Post, March 2012

10 F1: Trace a Single Lap

We can trace a single lap to display the speed (y-axis) coloured by gear as the vehicle travels around the circuit:

```
ggplot(subset(f1, Lap==2), aes(sLap, vCar)) +
  geom_line(aes(colour=NGear))
```

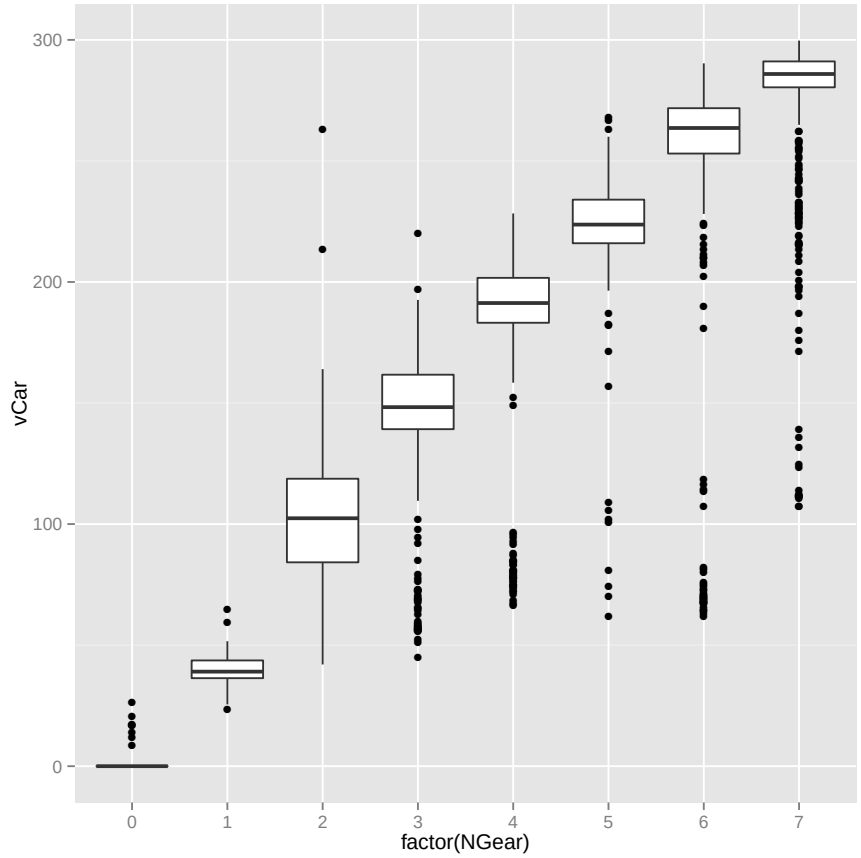


Based on Tony Hirst's Blog Post, March 2012

11 F1: Box Plot of Speed by Gear

Statistical graphics provide important insights. The box plot here makes sense, in that higher gears correspond to higher speeds.

```
ggplot(f1, aes(factor(NGear), vCar)) +
  geom_boxplot()
```

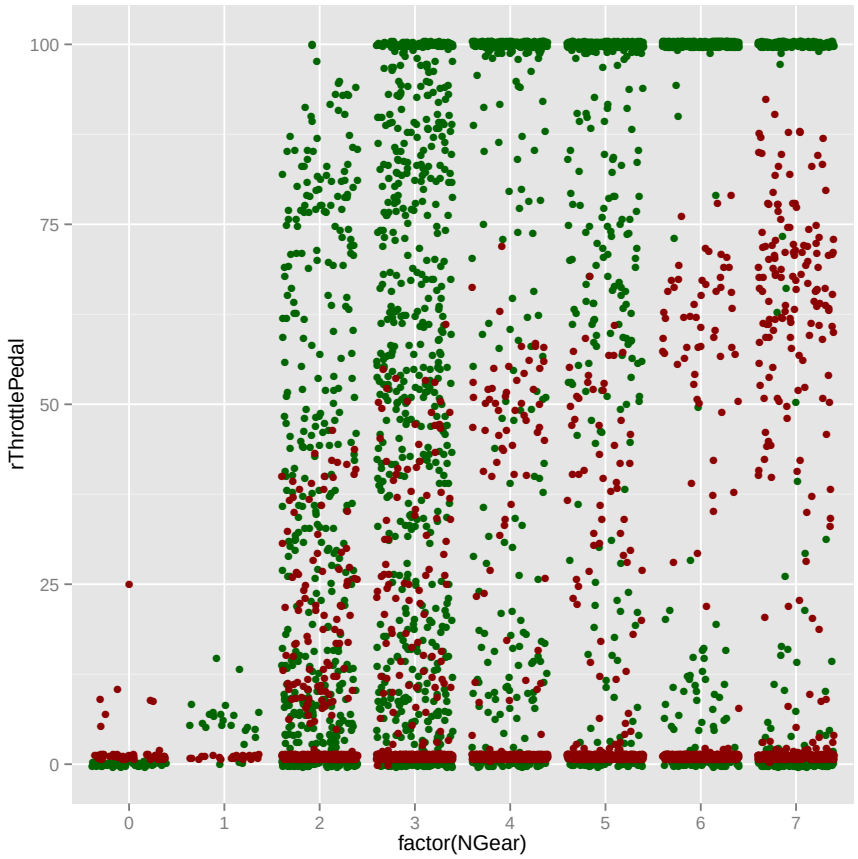


Based on Tony Hirst's Blog Post, March 2012

12 F1: Footwork

How busy are the feet? We can summarise the brake (red) and throttle (green) depression based on gear.

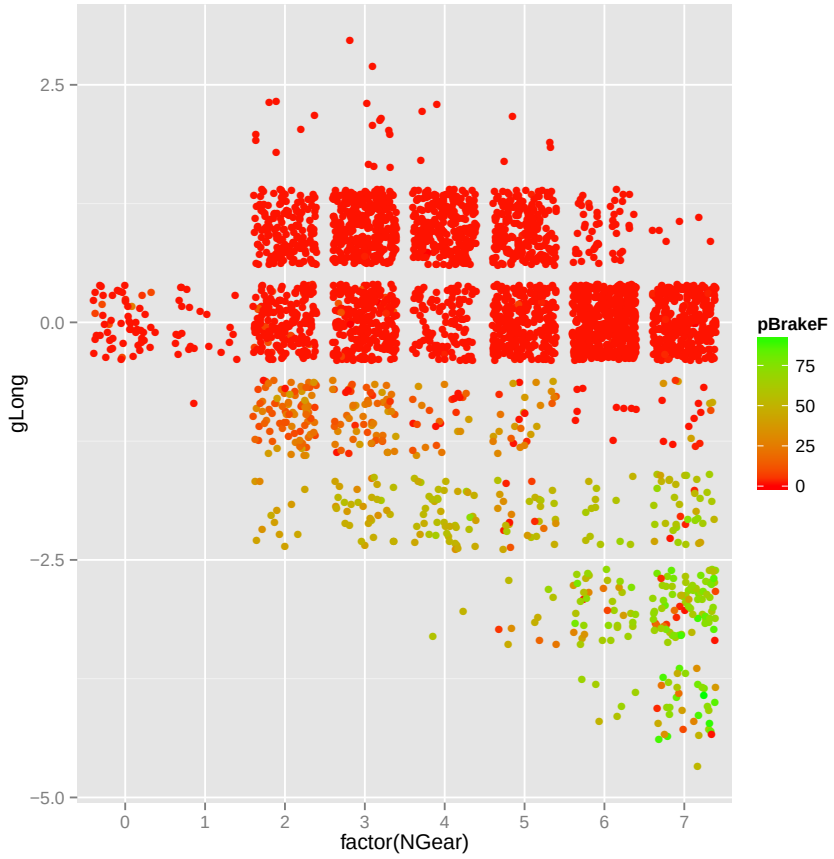
```
ggplot(f1, aes(factor(NGear))) +
  geom_jitter(aes(y=rThrottlePedal), colour='darkgreen') +
  geom_jitter(aes(y=pBrakeF), colour='darkred')
```



Based on Tony Hirst's Blog Post, March 2012

13 F1: Forces on the Driver

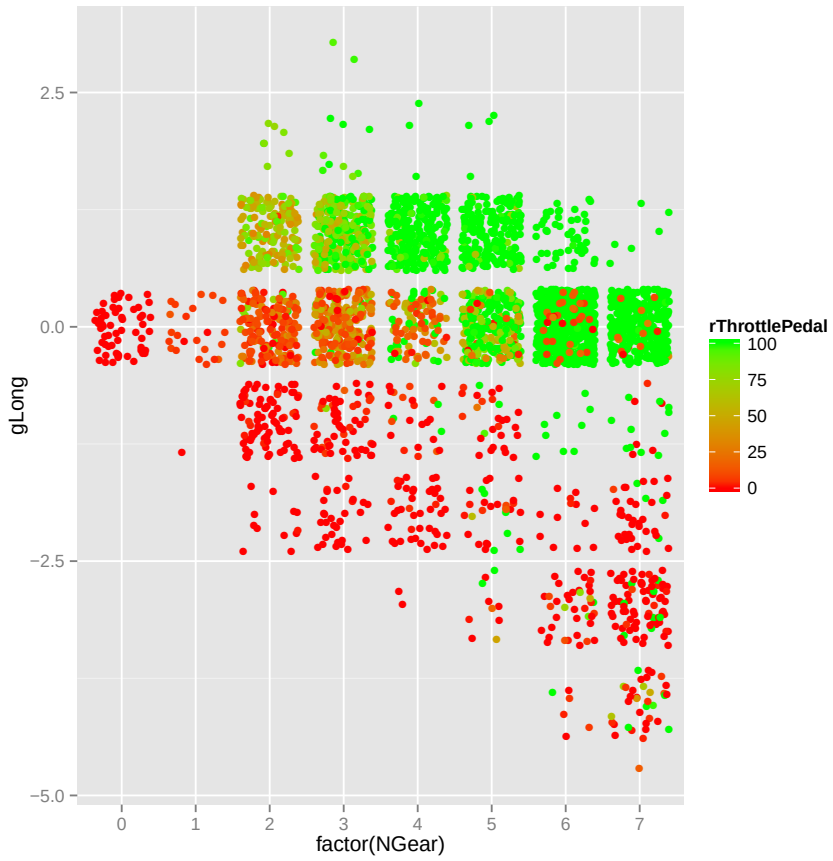
```
ggplot(f1, aes(factor(NGear), gLong)) +
  geom_jitter(aes(col=pBrakeF)) +
  scale_colour_gradient(low='red', high='green')
```



Based on Tony Hirst's Blog Post, March 2012

14 F1: More Forces

```
ggplot(f1, aes(factor(NGear), gLong)) +
  geom_jitter(aes(col=rThrottlePedal)) +
  scale_colour_gradient(low='red', high="green")
```

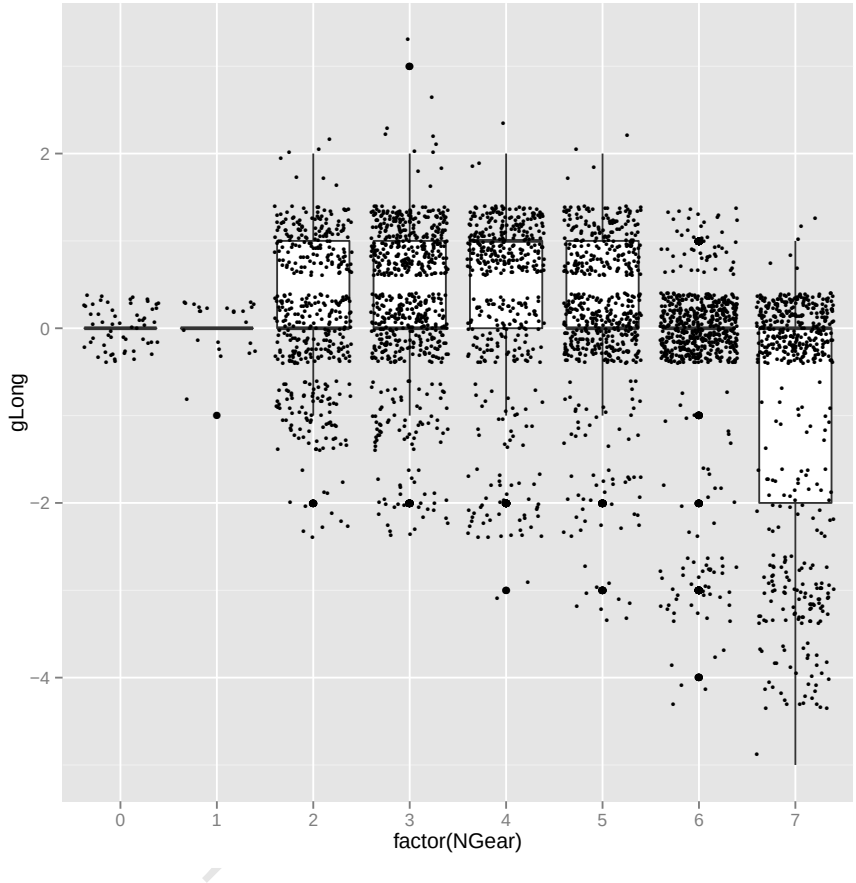


Based on Tony Hirst's Blog Post, March 2012

15 F1: Box Plot of Forces

We can use a box plot to investigate the longitudinal g-force's relationship with acceleration or braking by gear. Note that a random jitter is used to scatter points around their actual integer values.

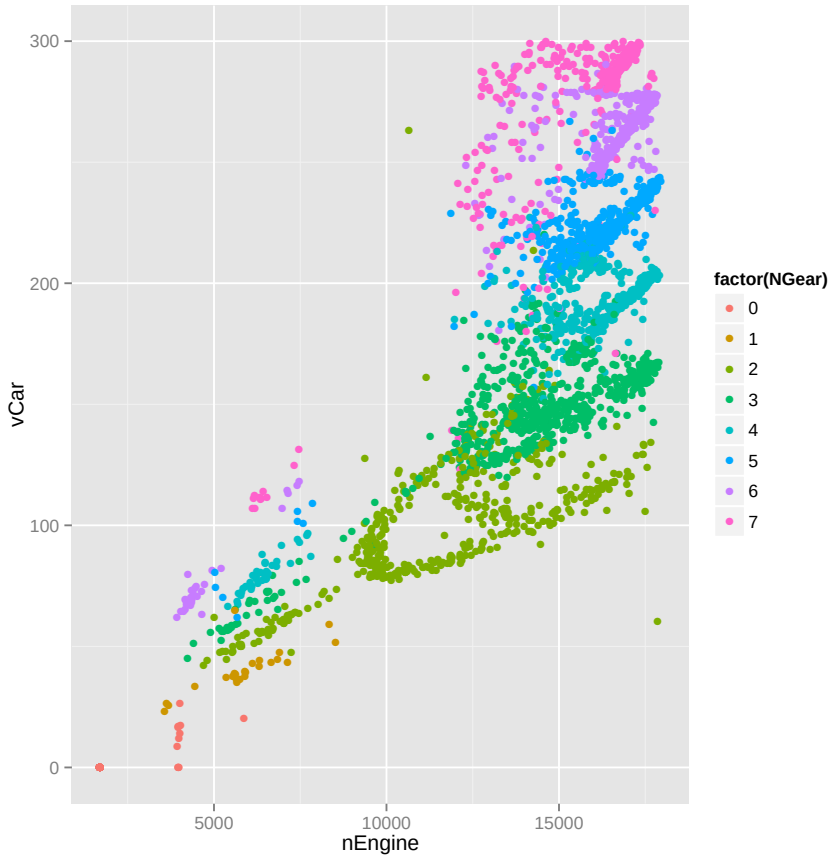
```
ggplot(f1, aes(factor(NGear), gLong)) +
  geom_boxplot() +
  geom_jitter(size=1)
```



Based on Tony Hirst's Blog Post, March 2012

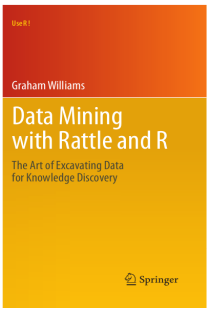
16 F1: RPM and Speed in Relation to Gear

```
ggplot(f1, aes(nEngine, vCar)) +
  geom_point(aes(col=factor(NGear)))
```



Based on Tony Hirst's Blog Post, March 2012

17 Further Reading and Acknowledgements



The Rattle Book, published by Springer, provides a comprehensive introduction to data mining and analytics using Rattle and R. It is available from Amazon. Other documentation on a broader selection of R topics of relevance to the data scientist is freely available from `http://datamining.togaware.com`, including the Datamining Desktop Survival Guide.

This chapter is one of many chapters available from `http://HandsOnDataScience.com`. In particular follow the links on the website with a * which indicates the generally more developed chapters.

Other resources include:

18 References

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Wickham H, Chang W (2014). *ggplot2: An implementation of the Grammar of Graphics*. R package version 1.0.0, URL http://CRAN.R-project.org/package=ggplot2.

Williams GJ (2009). "Rattle: A Data Mining GUI for R." *The R Journal*, **1**(2), 45–55. URL http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf.

Williams GJ (2011). *Data Mining with Rattle and R: The art of excavating data for knowledge discovery*. Use R! Springer, New York. URL http://www.amazon.com/gp/product/1441998896/ref=as_li_qf_sp_asin_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896.

This document, sourced from PlotsO.Rnw revision 509, was processed by KnitR version 1.6 of 2014-05-24 and took 6.8 seconds to process. It was generated by gjw on nyx running Ubuntu 14.04.1 LTS with Intel(R) Xeon(R) CPU W3520 @ 2.67GHz having 4 cores and 12.3GB of RAM. It completed the processing 2014-08-26 20:45:20.