Edwin Chen

Math, stats, machine learning, human computation, visualization, data science.

Previous:

- Math, linguistics at MIT
- Speech recognition at MSR
- Quant trading at Clarium Capital
- Ads quality at Twitter
- Quantitative analysis at Google

Email
Twitter
Github
Google+
LinkedIn
Quora

Atom / RSS

Recent Posts

Moving Beyond CTR: Better Recommendations Through Human Evaluation

Quick Introduction to ggplot2

by Edwin Chen
on Tue 17 January 2012

This is a bare-bones introduction to ggplot2, a visualization package in R. It assumes no knowledge of R.

For a better-looking version of this post, see this Github repository, which also contains some of the example datasets I use and a literate programming version of this tutorial.

Preview

Let's start with a preview of what ggplot2 can do.

Given Fisher's iris data set and one simple command…

```
1       qplot(Sepal.Length, Pe
```

…we can produce this plot of sepal length vs. petal length, colored by

Propensity Modeling, Causal Inference, and Discovering Drivers of Growth

Improving Twitter Search with Real-Time Human Computation

Edge Prediction in a Social Graph: My Solution to Facebook's User Recommendation Contest on Kaggle

Soda vs. Pop with Twitter

Infinite Mixture Models with Nonparametric Bayes and the Dirichlet Process

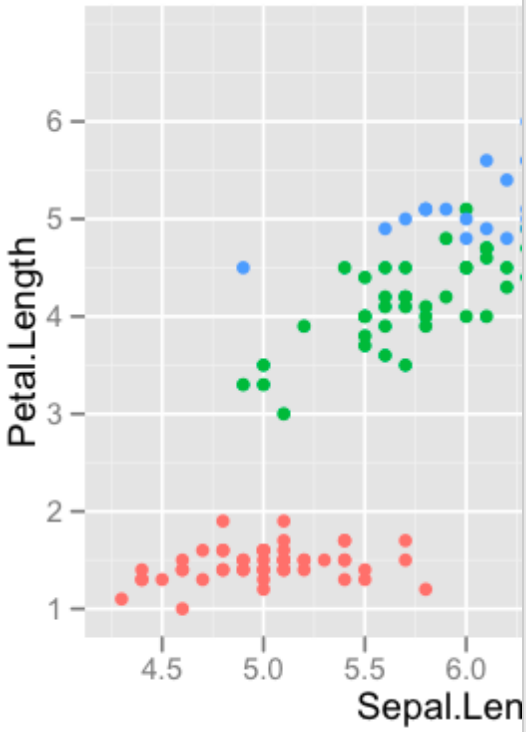
Instant Interactive Visualization with d3 + ggplot2

Movie Recommendations and More via MapReduce and Scalding

Quick Introduction to ggplot2

Introduction to Conditional Random Fields

species.



Installation

You can download R here. After installation, you can launch R in interactive mode by either typing R on the command line or opening the standard GUI (which should have been included in the download).

R Basics

Vectors

Vectors are a core data structure in R, and are created with `c()`. Elements in a vector must be of the same type.

```
numbers = c(23, 13, 5
  names = c("edwin",
```

Elements are indexed starting at 1, and are accessed with `[]` notation.

```
numbers[1] # 23
  names[1] # edwin
```

Data frames

Data frames are like matrices, but with named columns of different types (similar to database tables).

```
books = data.frame(
    title = c("harry
    author = c("rowli
    num_pages = c("35
  )
```

You can access columns of a data frame with `$`.

```
books$title # c("harr
  books$author[1] # "
```

You can also create new columns

with $.

```
books$num_bought_toda
  books$num_bought_ye

  books$total\_num\_b
```

read.table

Suppose you want to import a TSV file into R as a data frame.

tsv file without header

For example, consider the `data/students.tsv` file (with columns describing each student's age, test score, and name).

```
13   100 alice
  14   95  bob
  13   82  eve
```

We can import this file into R using `read.table()`.

```
students = read.table
    header = F, # fil
    sep = "\t", # fil
    col.names = c("ag
  )
```

We can now access the different columns in the data frame with `students$age`, `students$score`, and `students$name`.

csv file with header

For an example of a file in a different format, look at the `data/studentsWithHeader.tsv` file.

```
1       age,score,name
2         13,100,alice
3         14,95,bob
4         13,82,eve
```

Here we have the same data, but now the file is comma-delimited and contains a header. We can import this file with

```
1       students = read.table
2           sep = ",",
3           header = T  # fir
4         )
```

(Note: there is also a `read.csv` function that uses `sep = ","` by default.)

help

There are many more options that `read.table` can take. For a list of

these, just type `help(read.table)` (or `?read.table`) at the prompt to access documentation.

```
# These work for othe
help(read.table)
?read.table
```

ggplot2

With these R basics in place, let's dive into the ggplot2 package.

Installation

One of R's greatest strengths is its excellent set of packages. To install a package, you can use the `install.packages()` function.

```
install.packages("ggpl
```

To load a package into your current R session, use `library()`.

```
library(ggplot2)
```

Scatterplots with

qplot()

Let's look at how to create a scatterplot in ggplot2. We'll use the `iris` data frame that's automatically loaded into R.

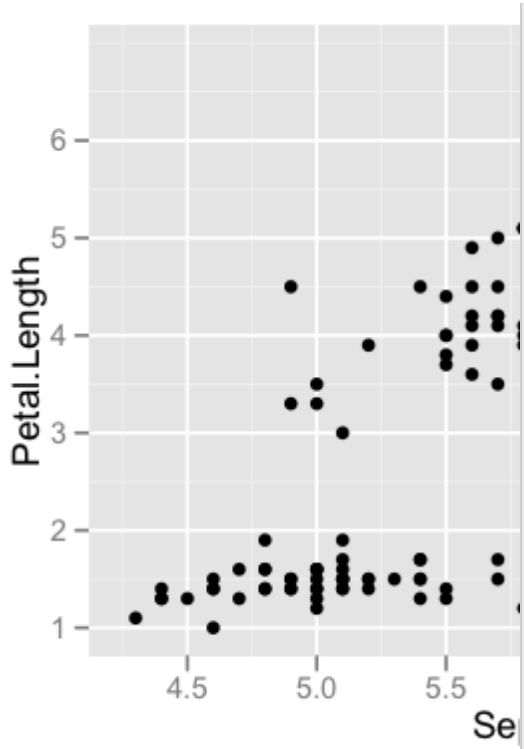
What does the data frame contain? We can use the `head` function to look at the first few rows.

```
1         head(iris) # by defa
2           head(iris, n = 10)
3
4           Sepal.Length Sepal
5                    5.1
6                    4.9
7                    4.7
8                    4.6
9                    5.0
10                   5.4
```

(The data frame actually contains three types of species: setosa, versicolor, and virginica.)

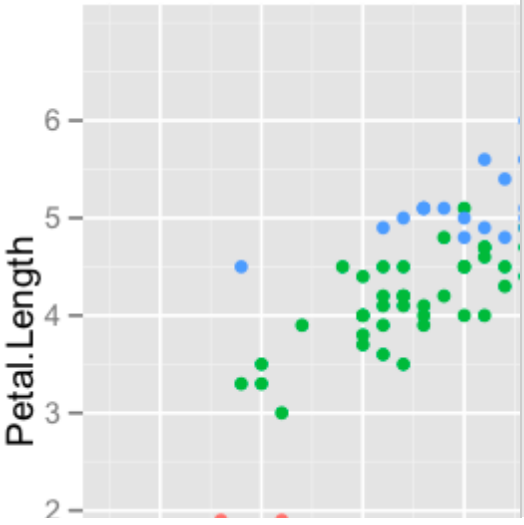
Let's plot `Sepal.Length` against `Petal.Length` using ggplot2's `qplot()` function.

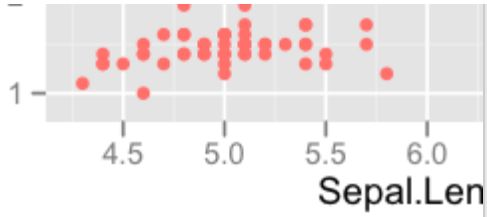
```
1       qplot(Sepal.Length, P
2         # Plot Sepal.Length
3         # * First argument
4         # * Second argument
5         # * `data = iris` m
```



To see where each species is located in this graph, we can color each point by adding a `color = Species` argument.

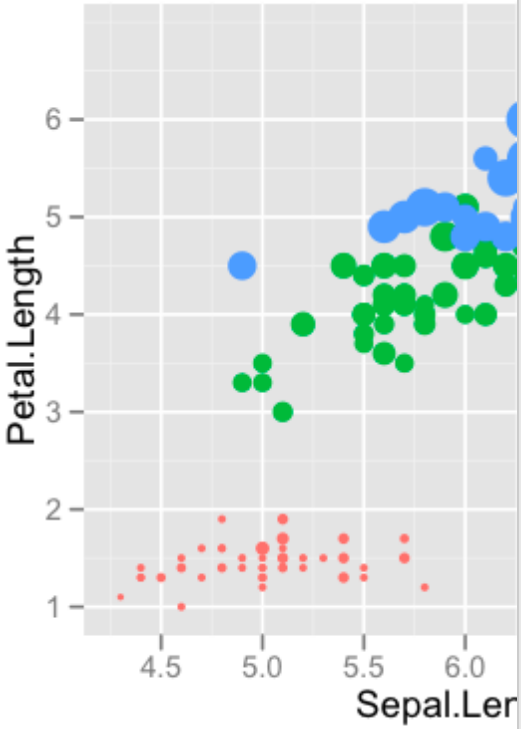
```
1      qplot(Sepal.Length, Pe
```



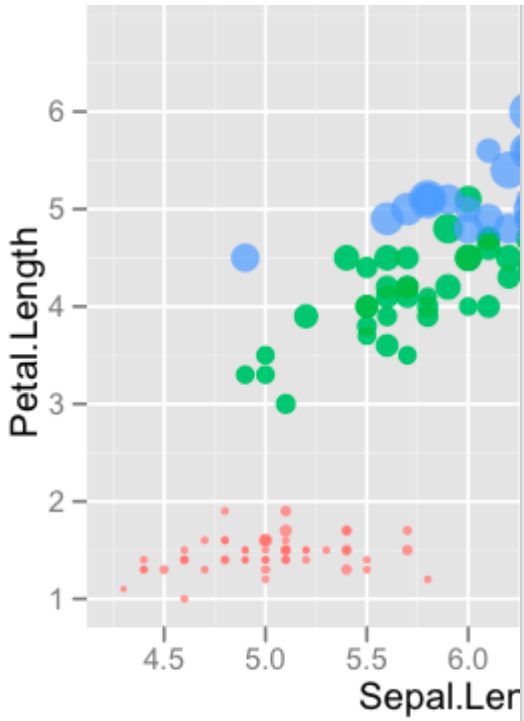


Similarly, we can let the size of each point denote sepal width, by adding a `size = Sepal.Width` argument.

```
1       qplot(Sepal.Length, P
2         # We see that Iris
```

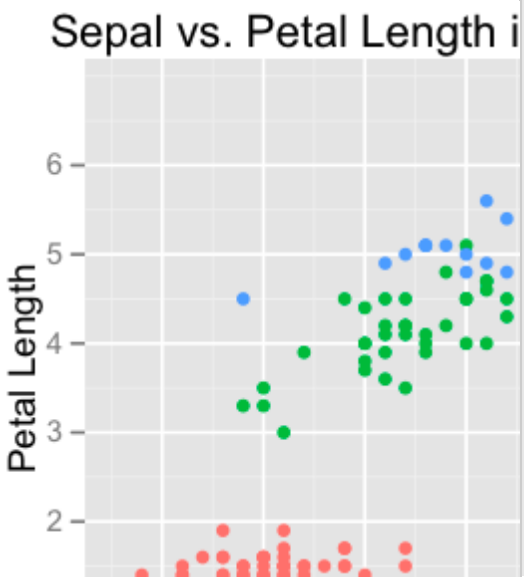


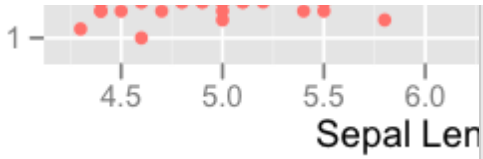

```
1       qplot(Sepal.Length, P
2         # By setting the al
```



Finally, let's fix the axis labels and add a title to the plot.

```
1       qplot(Sepal.Length, P
2           xlab = "Sepal Len
3           main = "Sepal vs.
```





Other common geoms

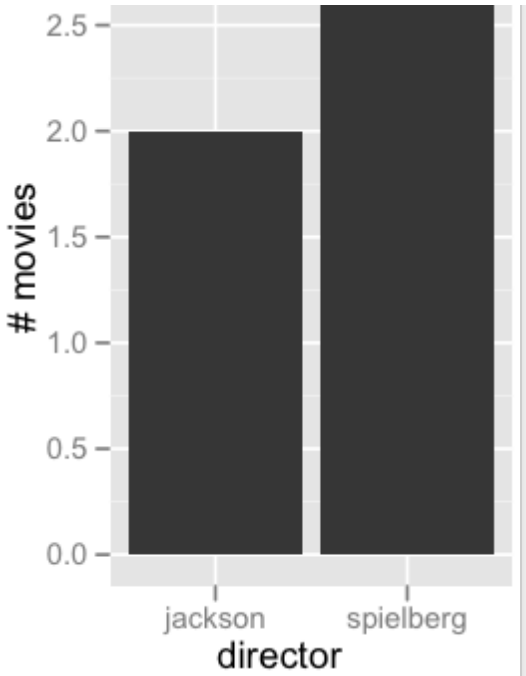
In the scatterplot examples above, we implicitly used a *point* **geom**, the default when you supply two arguments to `qplot()`.

```
# These two invocatio
  qplot(Sepal.Length,
  qplot(Sepal.Length,
```

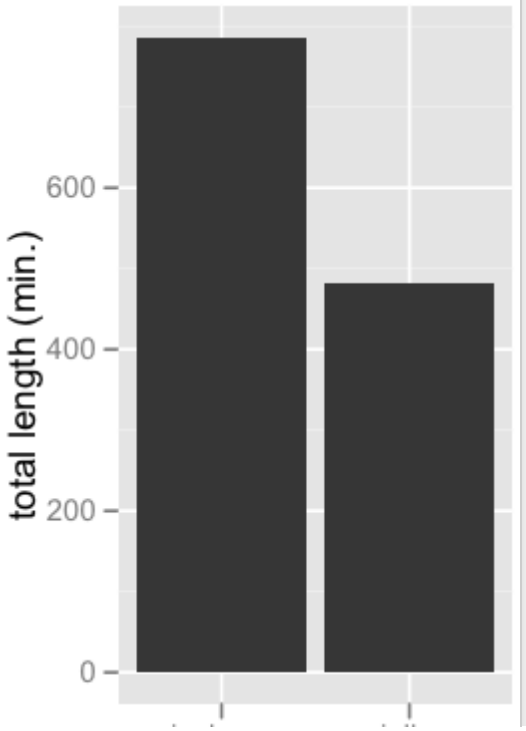
But we can also easily use other types of geoms to create more kinds of plots.

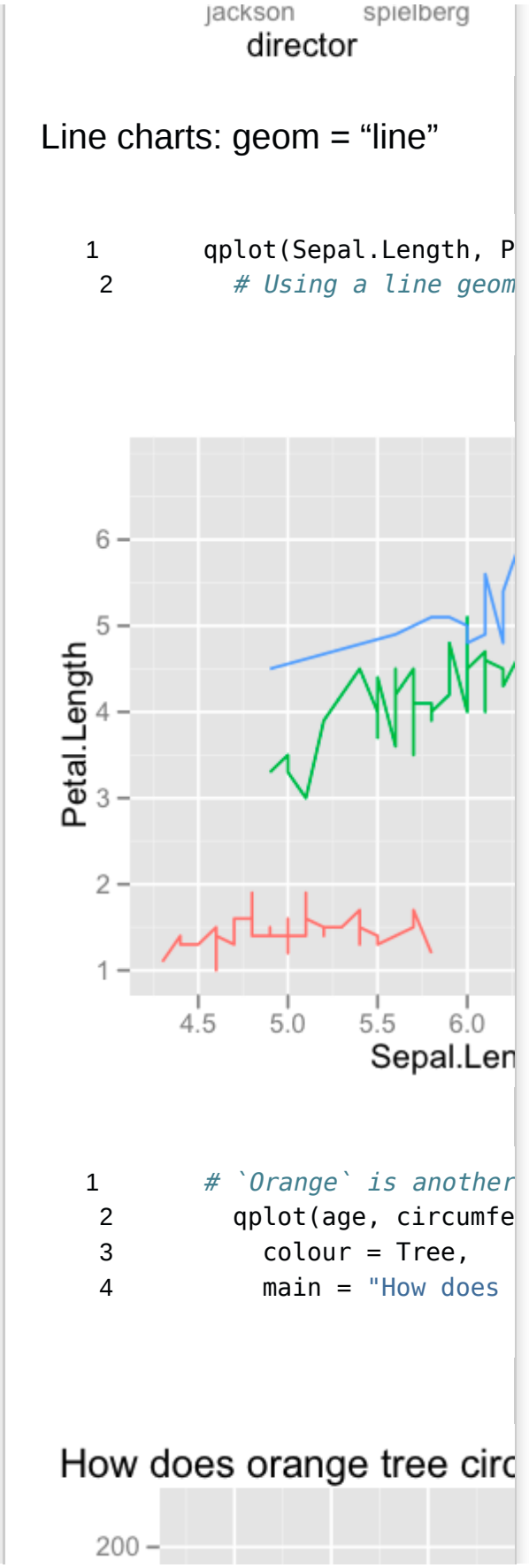
Barcharts: geom = “bar”

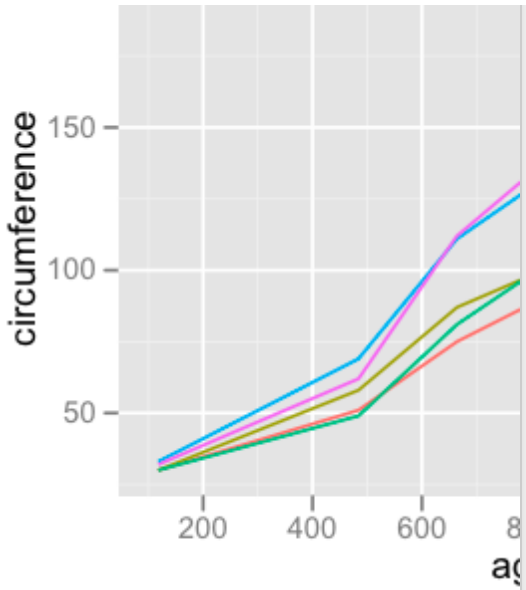
```
movies = data.frame(
    director = c("spi
    movie = c("jaws",
    minutes = c(124,
  )

  # Plot the number o
  qplot(director, dat
  # By default, the h
```



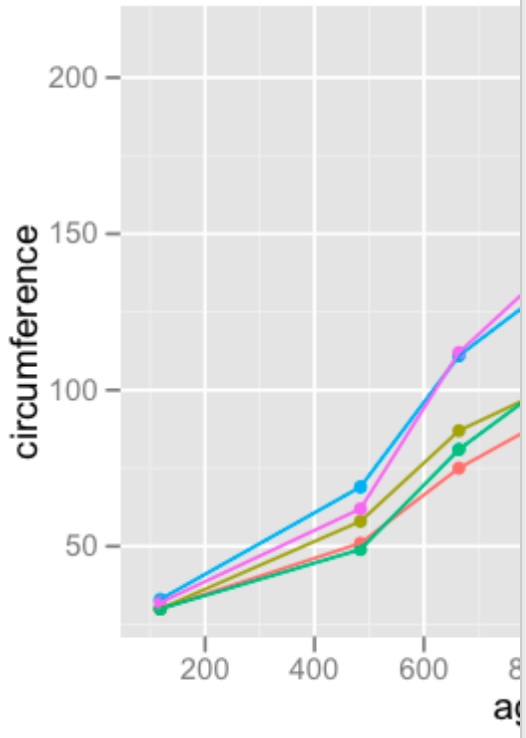

```
1       # But we can also sup
2         # Here the height o
3         qplot(director, wei
```




jackson
spielberg
director
Line charts: geom = “line”
1 qplot(Sepal.Length, P
2 # Using a line geom
6
5
4
3
2
1
Petal.Length
4.5
5.0
5.5
6.0
Sepal.Len
1 # `Orange` is another
2 qplot(age, circumfe
3 colour = Tree,
4 main = "How does
How does orange tree circ
200




```
1       # We can also plot bo
2         qplot(age, circumfe
```



And that's it with what I'll cover.

Next Steps

I skipped over a lot of aspects of R and ggplot2 in this intro.

For example,

- There are many geoms (and other functionalities) in ggplot2 that I didn't cover, e.g., boxplots and histograms.
- I didn't talk about ggplot2's layering system, or the grammar of graphics it's based on.

So I'll end with some additional resources on R and ggplot2.

- I don't use it myself, but RStudio is a popular IDE for R.
- The official ggplot2 documentation is great and has lots of examples. There's also an excellent book.
- plyr is another fantastic R package that's also by Hadley Wickham (the author of ggplot2).
- The official R introduction is okay, but definitely not great. I haven't found any R tutorials I really like, but I've heard good things about The Art of R Programming.