concordance=TRUE

Introduction to ggplot2

Mark Dunning

November 1, 2013

Package by Hadley Wickham

- **G**rammar of **G**raphics approach
- **'Base graphics are good for drawing pictures. ggplot2 graphics are good for understanding the data'** - Hadley Wickham
- Extensive online help, videos, and **google**

```
library(ggplot2)
```

The ggplot2 ethos

- A plot is made up of multiple layers
- A layer consists of *data*, a set of *mappings* between variables and aesthetics, a *geom*etric object and a *statistical* transformation
- *Scales* control the details of the mapping
- All components are independant and reusable
- Carefully chosen defaults
- Less time spent on making plot look good, more time for interpreting the data

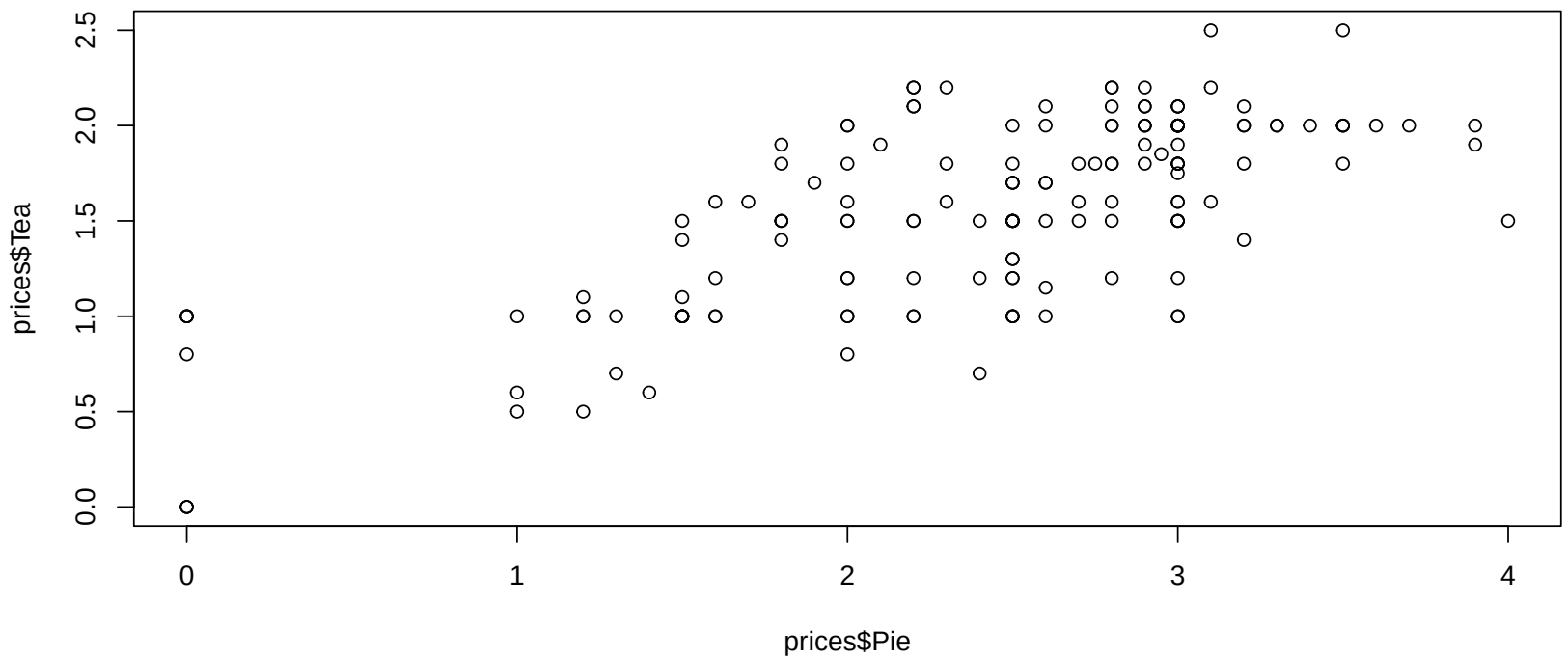
Load the data

```
prices <- read.csv("data/priceData.txt", sep = "\t")

head(prices, 2)

##          Club League Cheapest.season.ticket
## 1     Arsenal      1                    985
## 2 Aston Villa      1                    325
##   Most.expensive.season.ticket
## 1                         1955
## 2                          595
##   Cheapest.match.day.ticket
## 1                        26
## 2                        20
##   Most.expensive.match.day.ticket
## 1                             126
## 2                              45
##   Cheapest.day.out Programme Pie Tea   lat
## 1             34.3         3 3.3 2.0 51.55
## 2             28.3         3 3.2 2.1 52.51
##       lon      e      n
## 1 -0.1086 531225 185700
## 2 -1.8848 407919 290126
```

Lets explore the data

Suppose we are interested in the relationship between the price of tea, and the price of pies

```
plot(prices$Pie, prices$Tea)
```



Lets try in ggplot2

The equivalent of `plot` is `ggplot` which requires `data` and `aes` arguments
`aes` defines the aesthetic mappings to pass to the plot. `data` must be a data frame

```
ggplot(data = prices, aes(x = Pie, y = Tea))
```

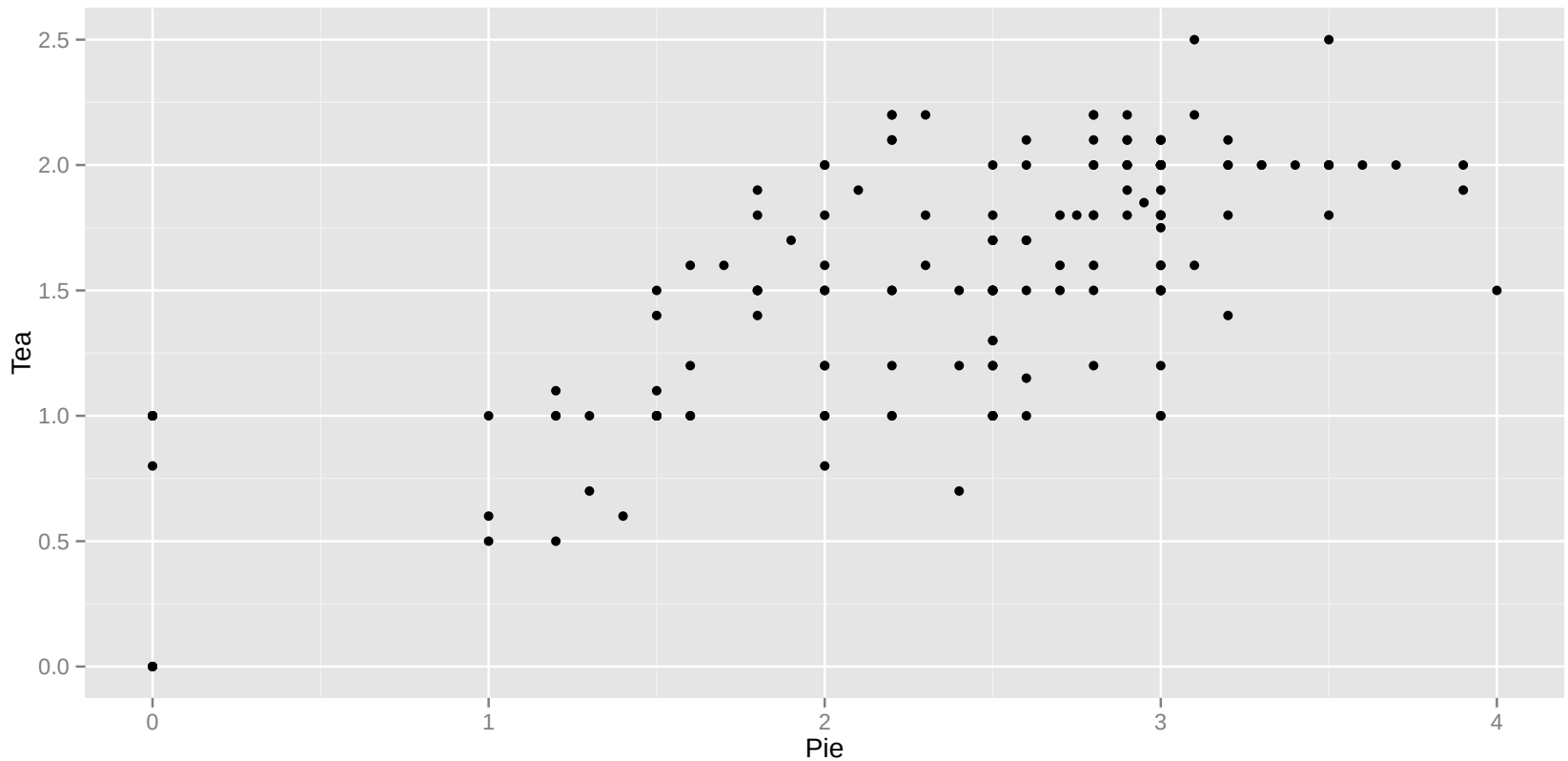
```
## Error:  No layers in plot
```

We haven't specified a *geom*

Pick a geom, any geom

We have to 'add' a layer to the plot

```
ggplot(prices, aes(x = Pie, y = Tea)) +
    geom_point()
```



The command in detail

```
ggplot(prices, aes(x = Pie, y = Tea)) +
    geom_point()
```

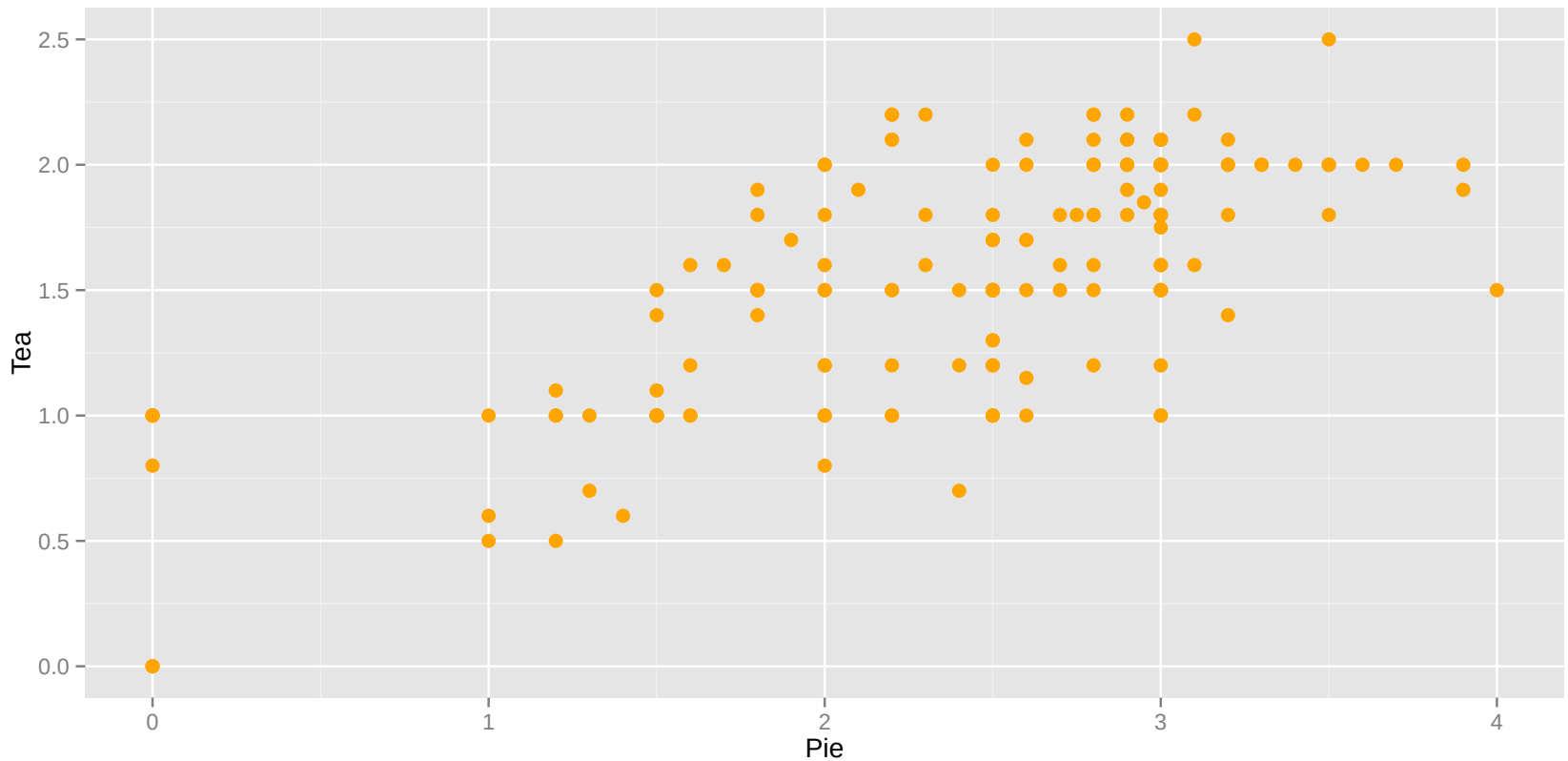
- Specify data and variable inside ggplot function
- Add layers of geometric objects, statistical models and panels
- geom_point knows about the data and aesthetics (it inherits them)

Adding color

geom_point

can have it's own set of aesthetics

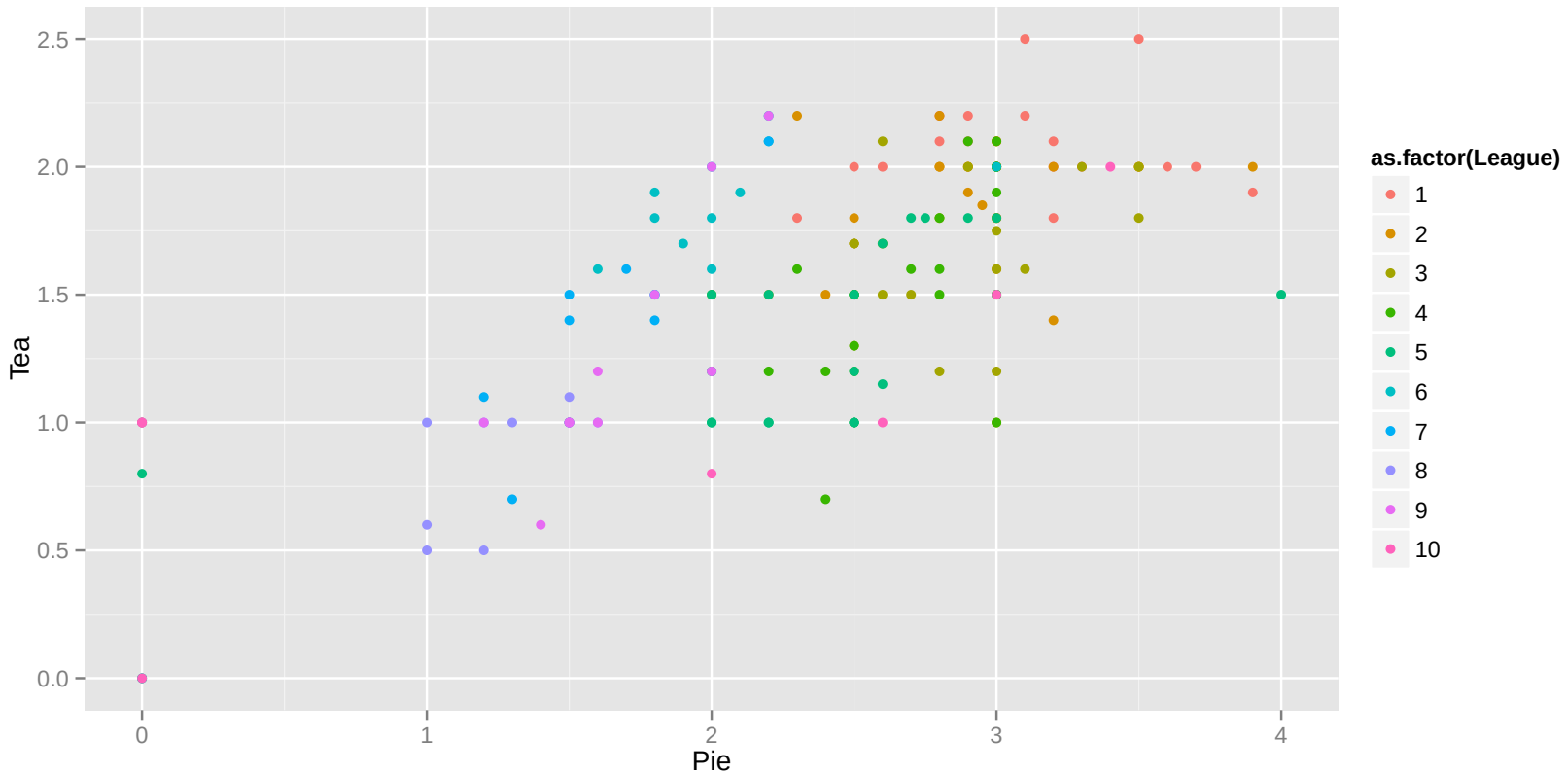
```
ggplot(prices, aes(x = Pie, y = Tea)) +
    geom_point(color = "orange",
        size = 3)
```



Adding color

Other aesthetics can be set in the `ggplot` call such as `colour`, `shape`, `size`. Legend is set automatically.

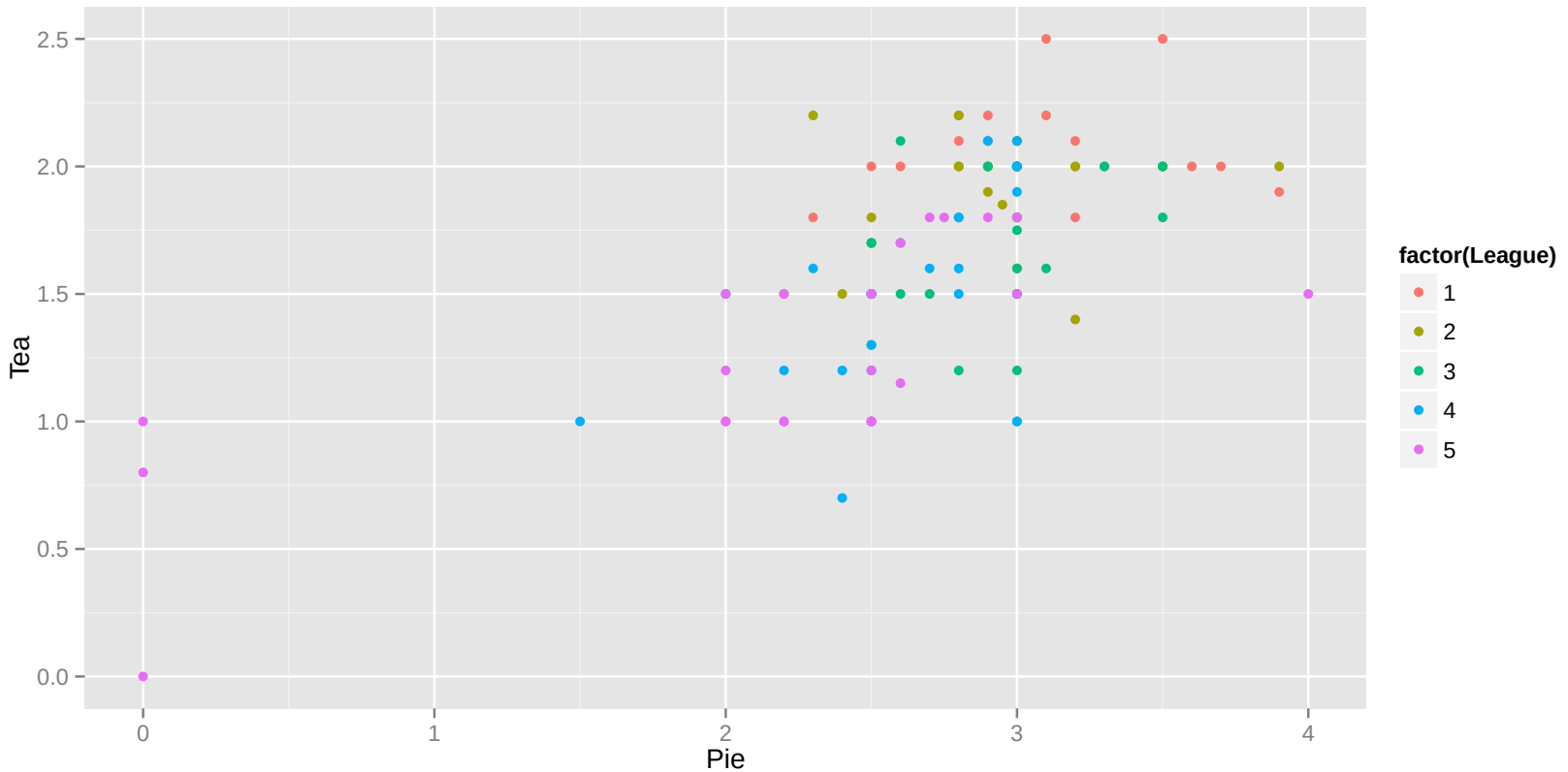
```
ggplot(prices, aes(x = Pie, y = Tea,
    color = as.factor(League))) +
    geom_point()
```



Adding color

Note that legend is automatically re-drawn

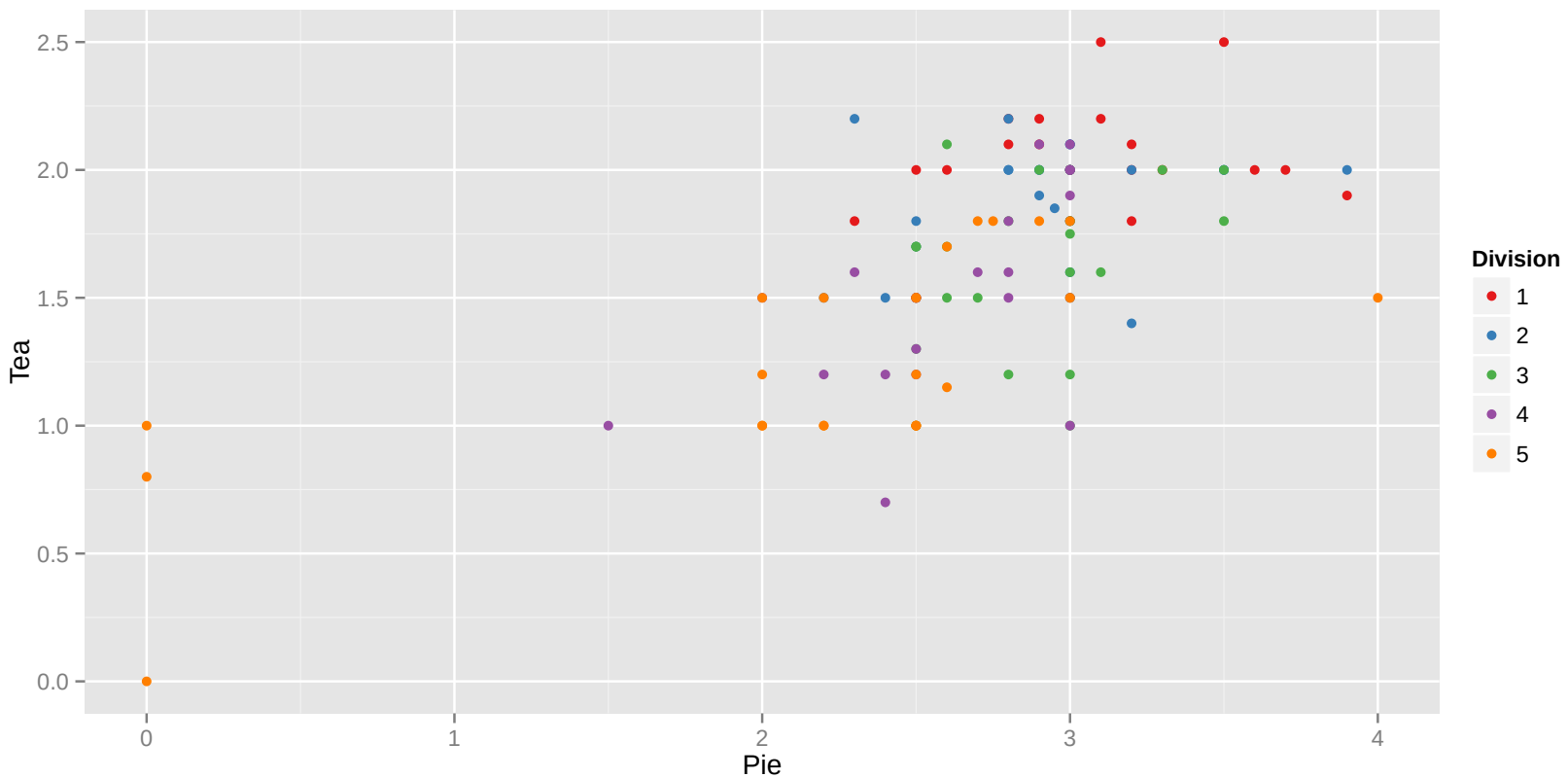
```
sub <- prices$League < 6
ggplot(prices[sub, ], aes(x = Pie,
    y = Tea, color = factor(League))) +
    geom_point()
```



Changing colours and legend

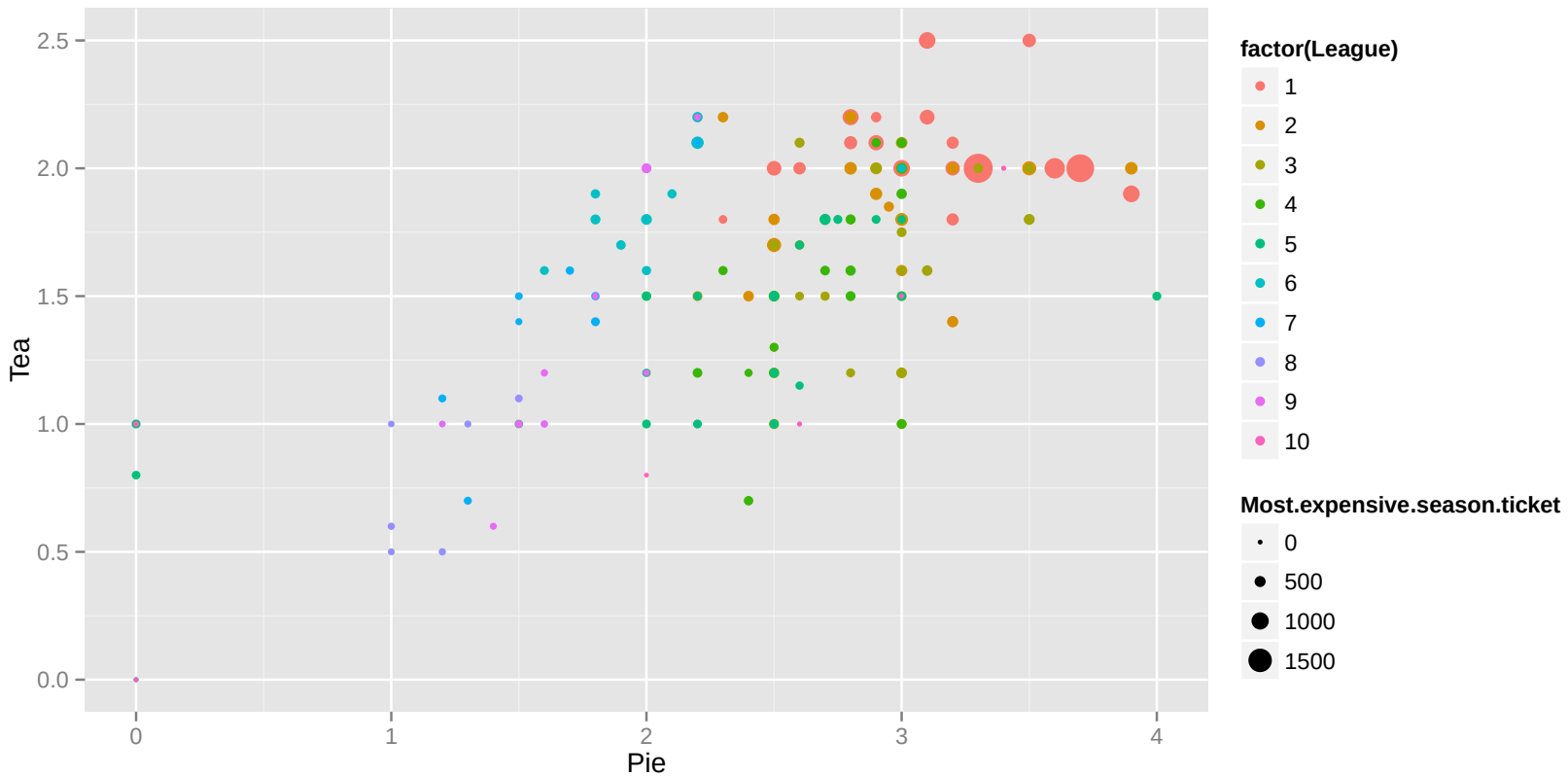
But we can specify colours and legend manually

```
sub <- prices$League < 6
library(RColorBrewer)
ggplot(prices[sub, ], aes(x = Pie,
    y = Tea, color = factor(League))) +
    geom_point() + scale_color_manual(name = "Division",
    values = brewer.pal(5, "Set1"))
```



Adding size

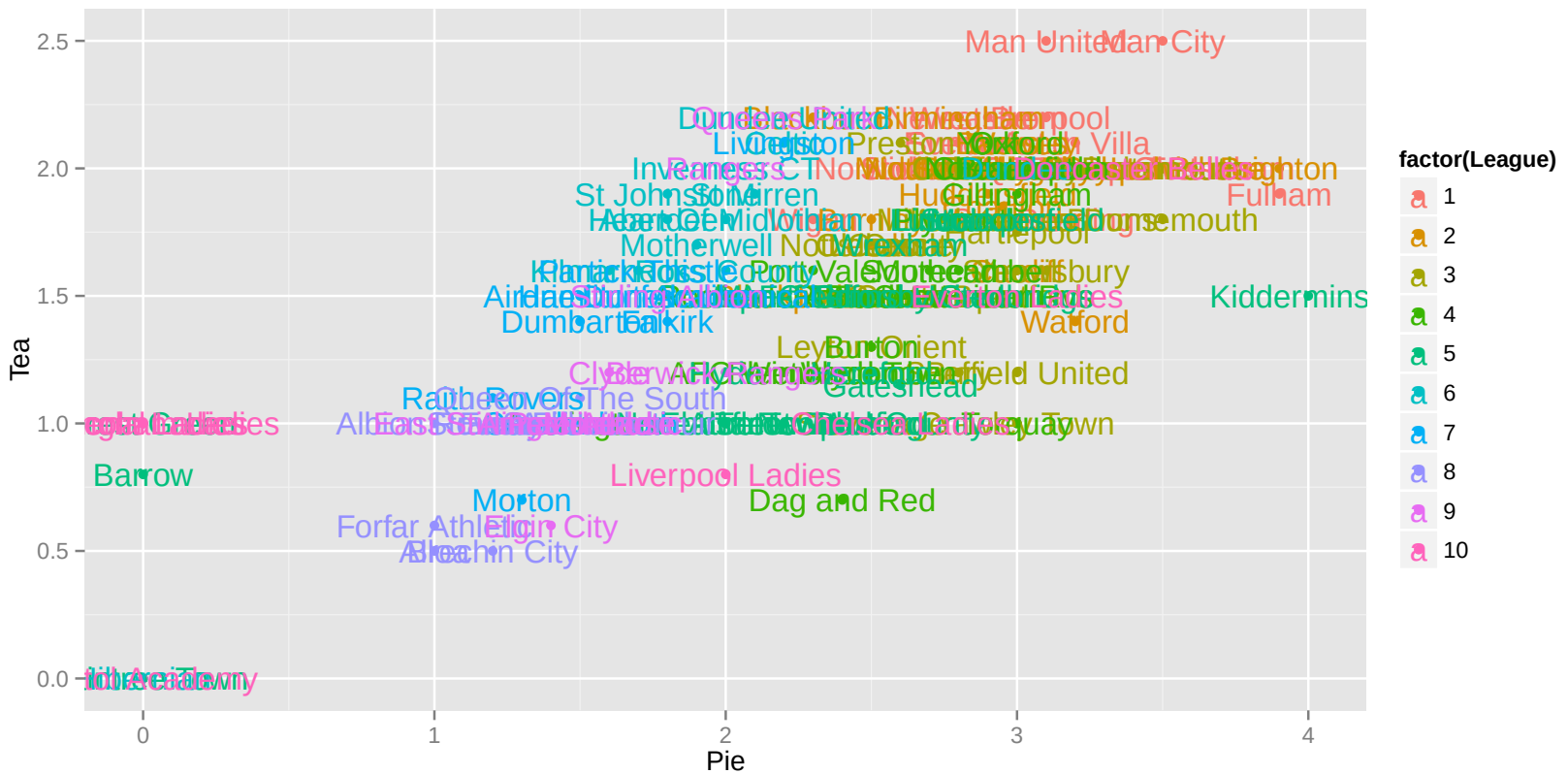
```
ggplot(prices, aes(x = Pie, y = Tea,
    color = factor(League), size = Most.expensive.season.ticket)) +
    geom_point()
```



Adding labels

For labels we need to add another 'layer' using `geom_text`. This requires a label aesthetic to be defined

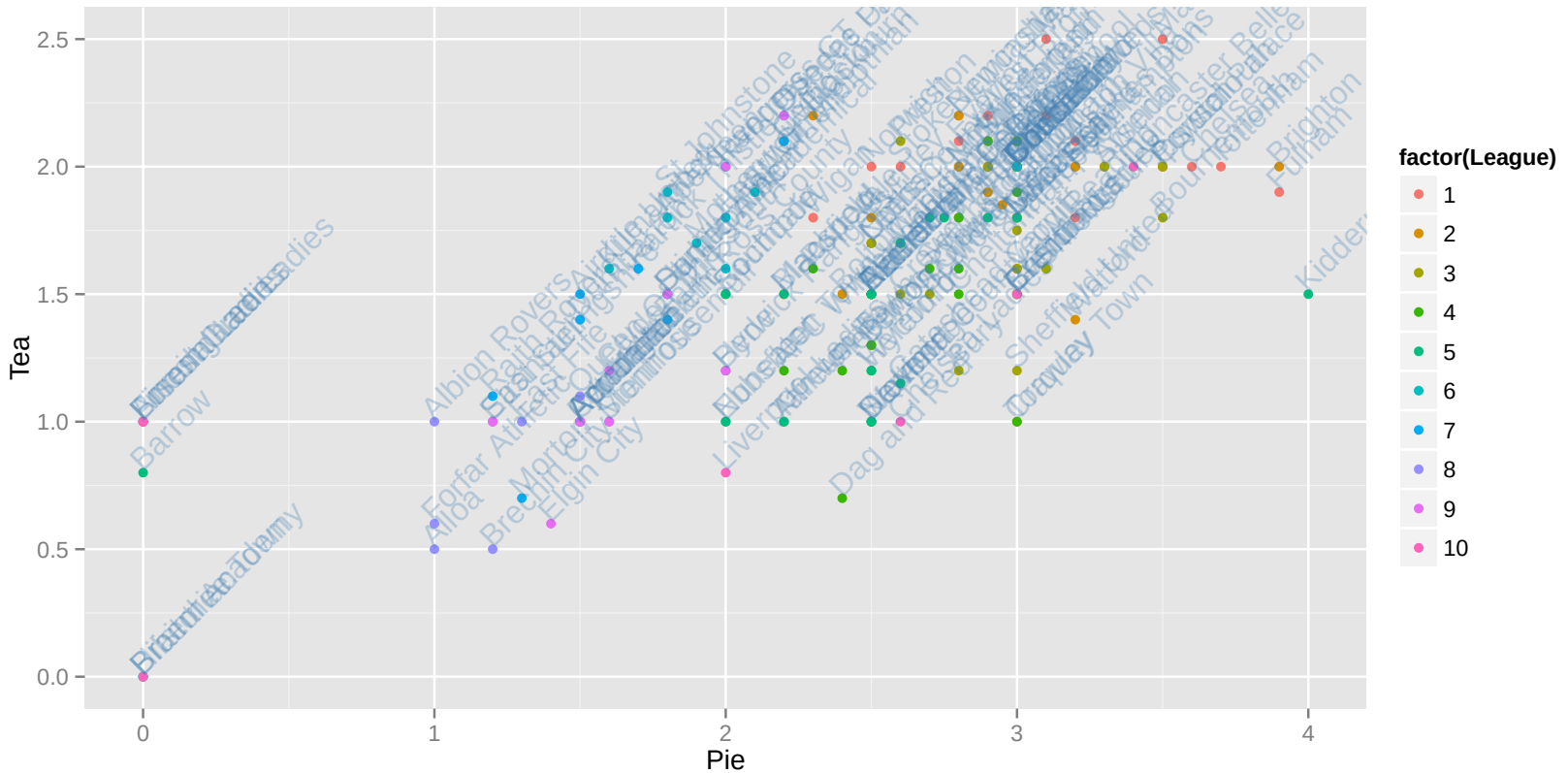
```
ggplot(prices, aes(x = Pie, y = Tea,
    color = factor(League), label = Club)) +
    geom_point() + geom_text()
```



Adding labels

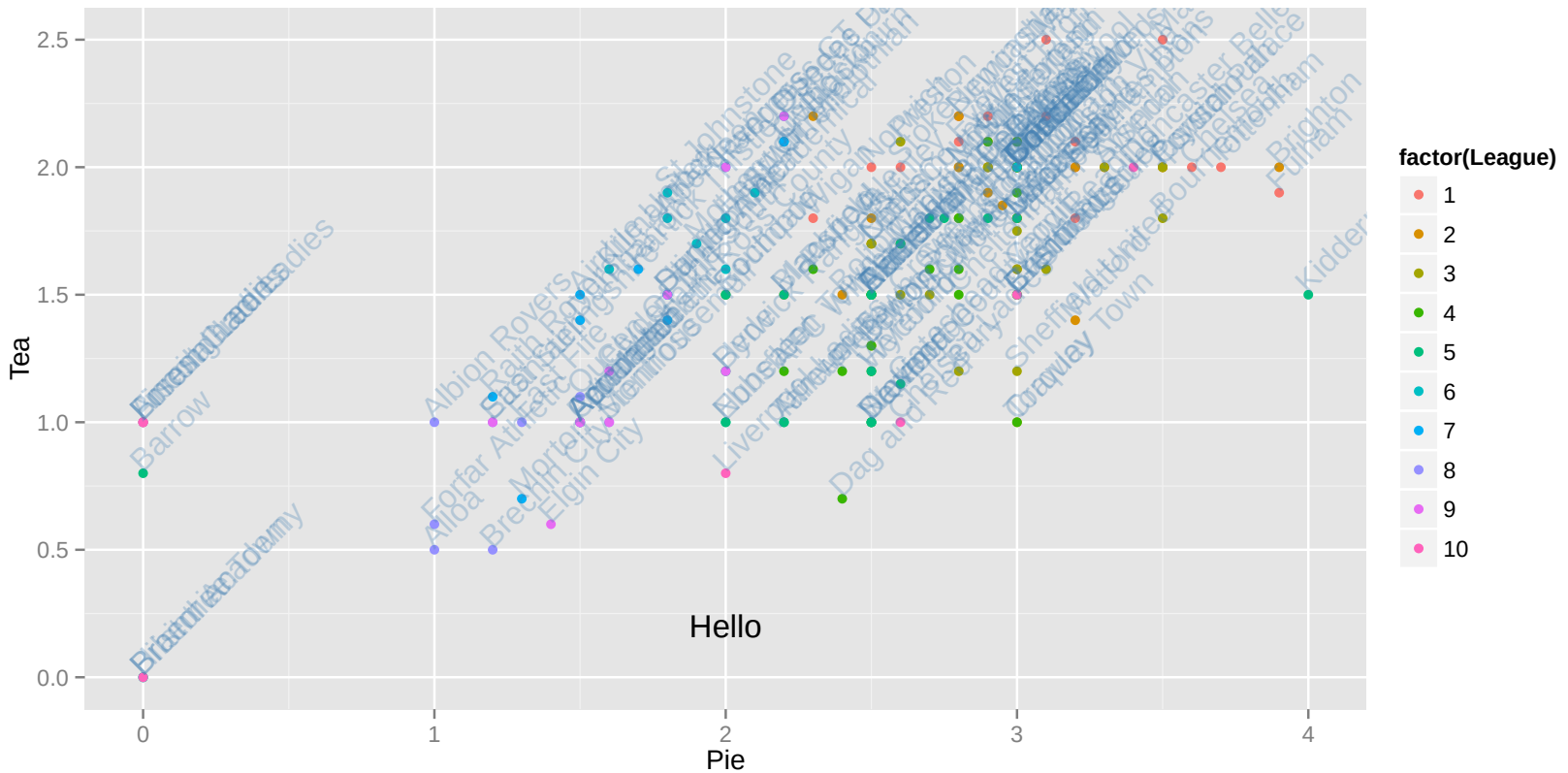
Can have more control over the labels

```
ggplot(prices, aes(x = Pie, y = Tea,
    color = factor(League), label = Club)) +
    geom_point() + geom_text(angle = 45,
    vjust = 0, hjust = 0, color = "steelblue",
    alpha = 0.3)
```



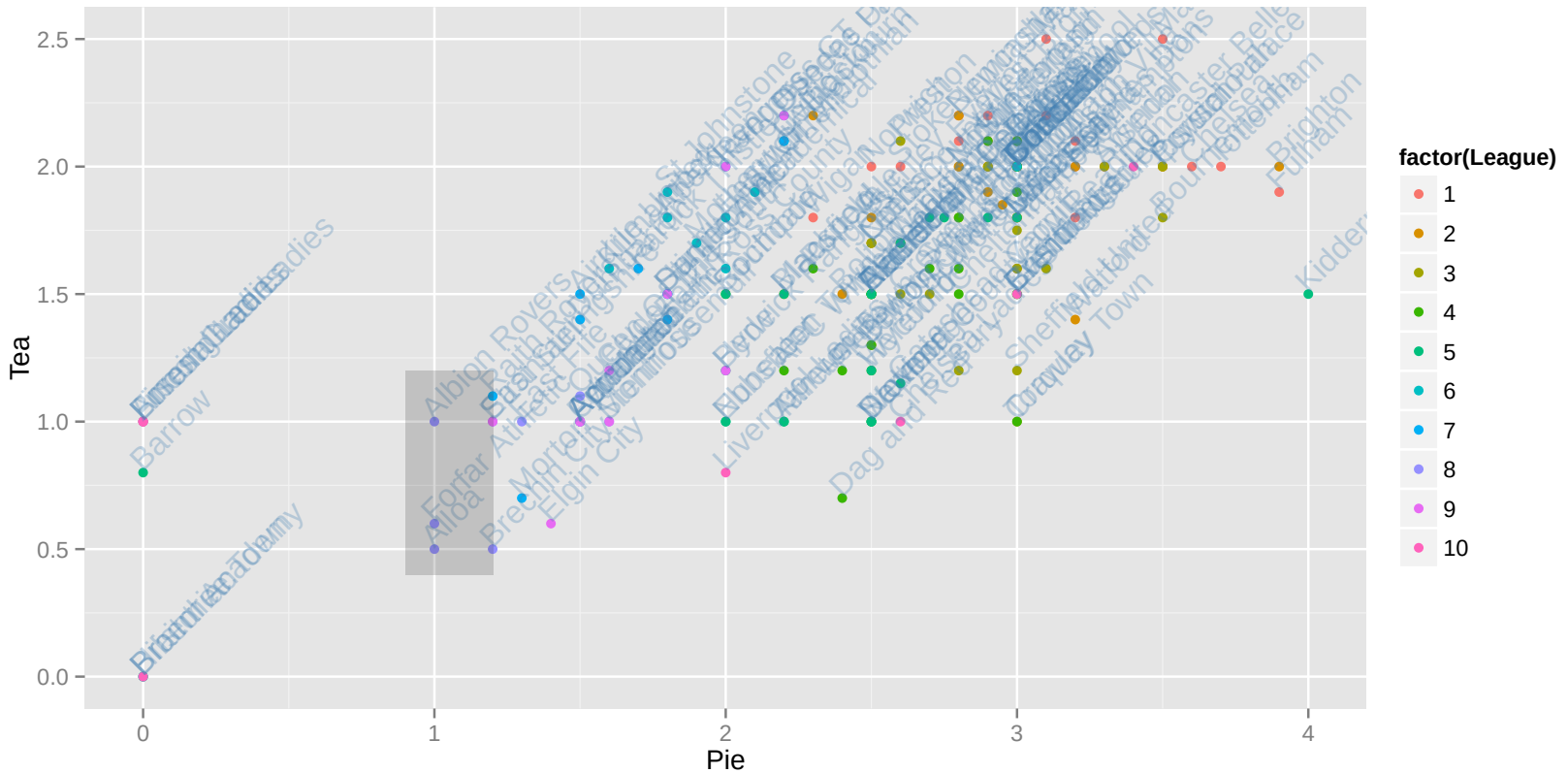
Annotating

```
ggplot(prices, aes(x = Pie, y = Tea,
    color = factor(League), label = Club)) +
    geom_point() + geom_text(angle = 45,
    vjust = 0, hjust = 0, color = "steelblue",
    alpha = 0.3) + annotate("text",
    2, 0.2, label = "Hello")
```



Annotating

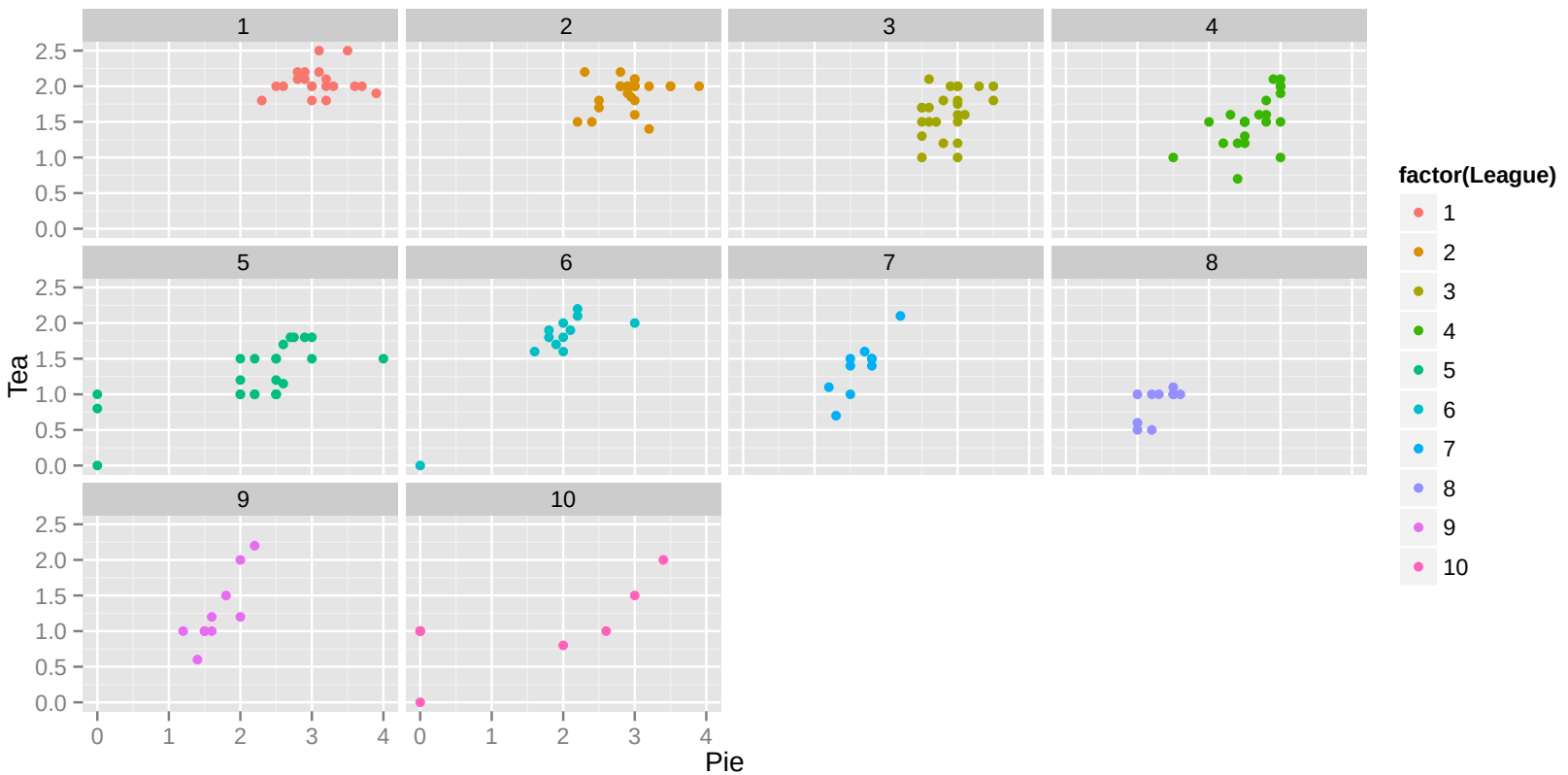
```
ggplot(prices, aes(x = Pie, y = Tea,
    color = factor(League), label = Club)) +
    geom_point() + geom_text(angle = 45,
    vjust = 0, hjust = 0, color = "steelblue",
    alpha = 0.3) + annotate("rect",
    xmin = 0.9, xmax = 1.2, ymin = 0.4,
    ymax = 1.2, alpha = 0.2)
```



Faceting

Faceting is an important tool to break the data into subsets for plotting.

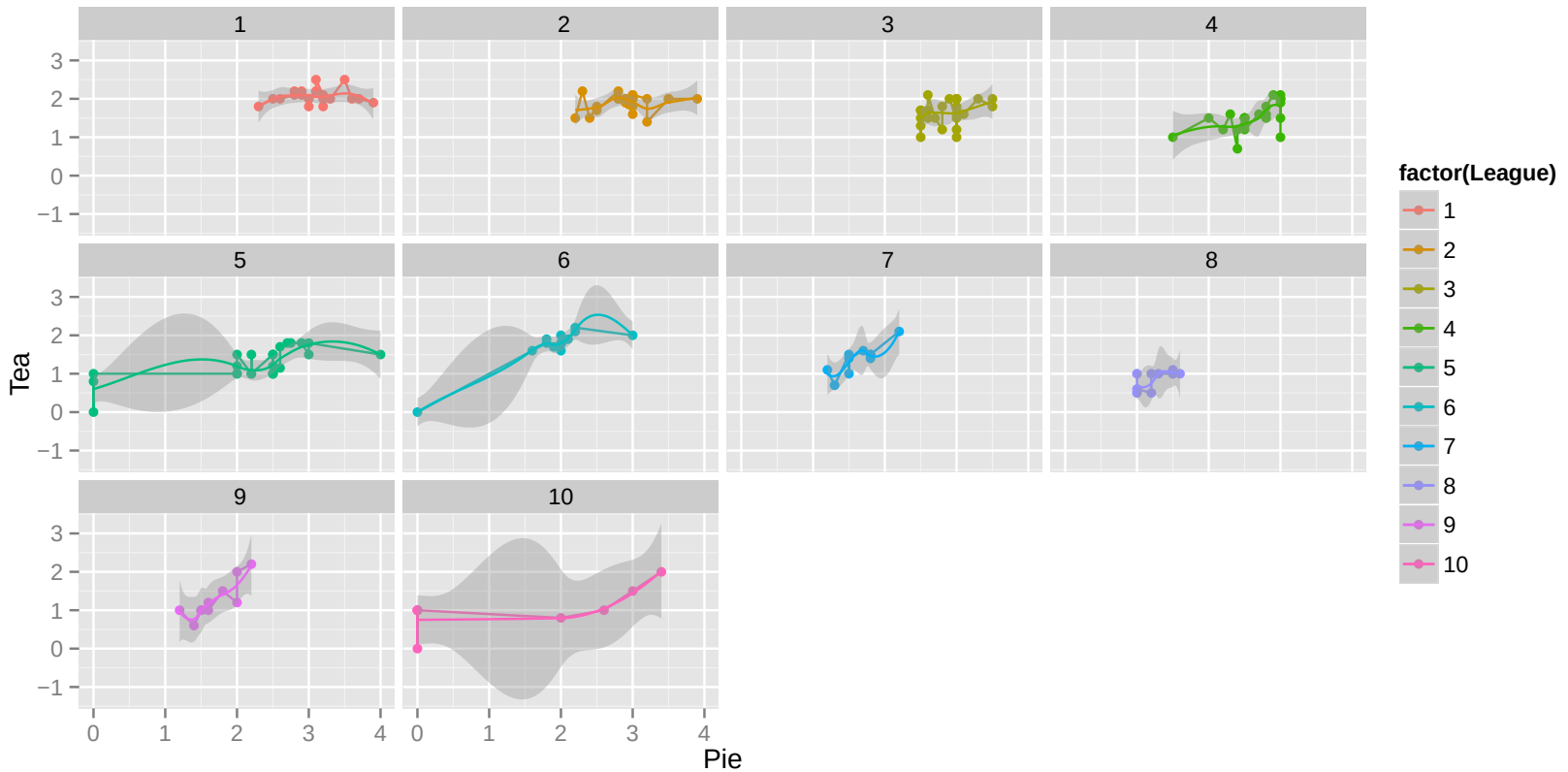
```
ggplot(prices, aes(x = Pie, y = Tea,
    color = factor(League))) +
    geom_point() + facet_wrap(~League)
```



Faceting

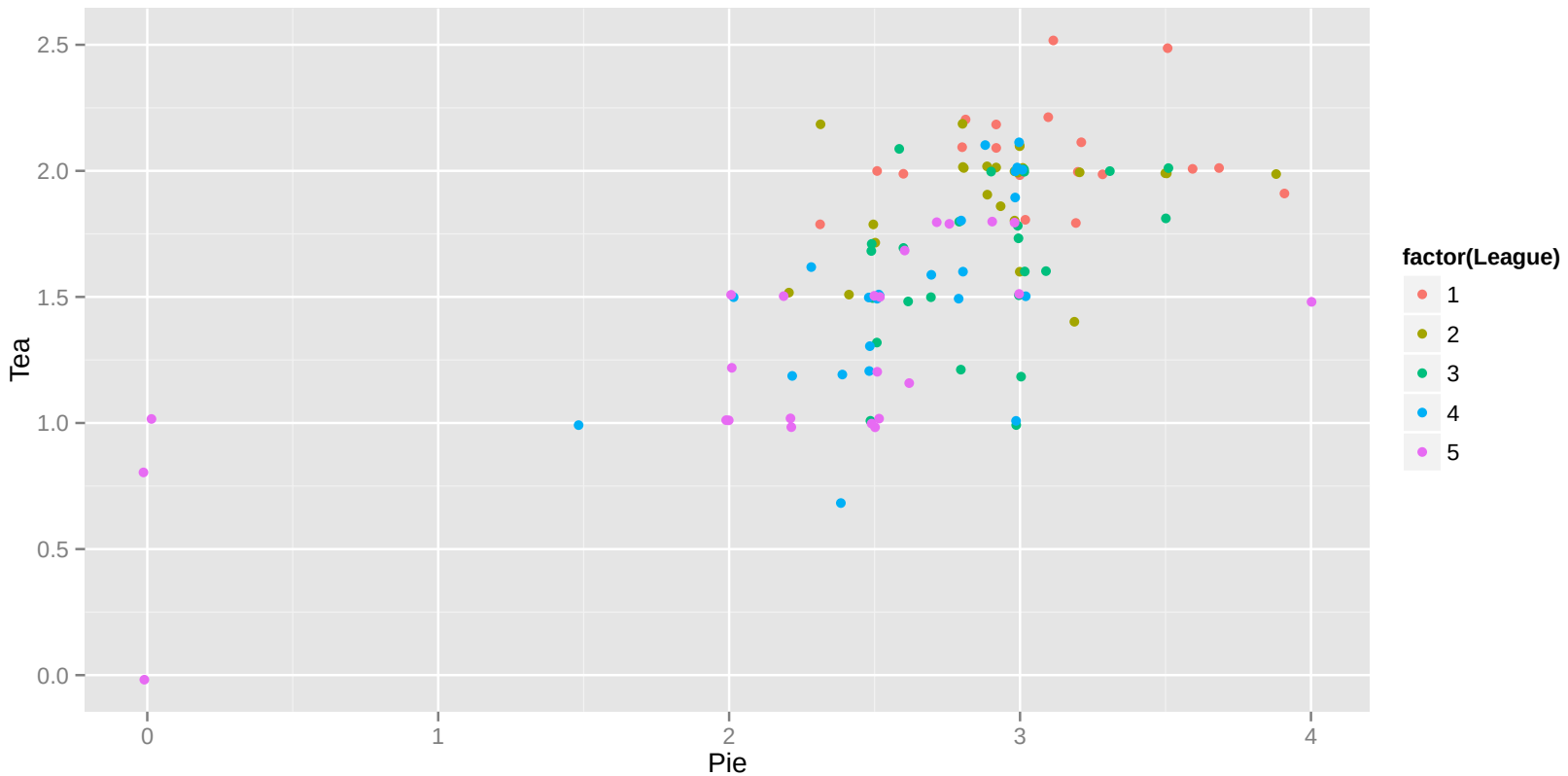
Lines and smoothing

```
ggplot(prices, aes(x = Pie, y = Tea,
    color = factor(League))) +
    geom_point() + geom_line() +
    geom_smooth() + facet_wrap(~League)
```



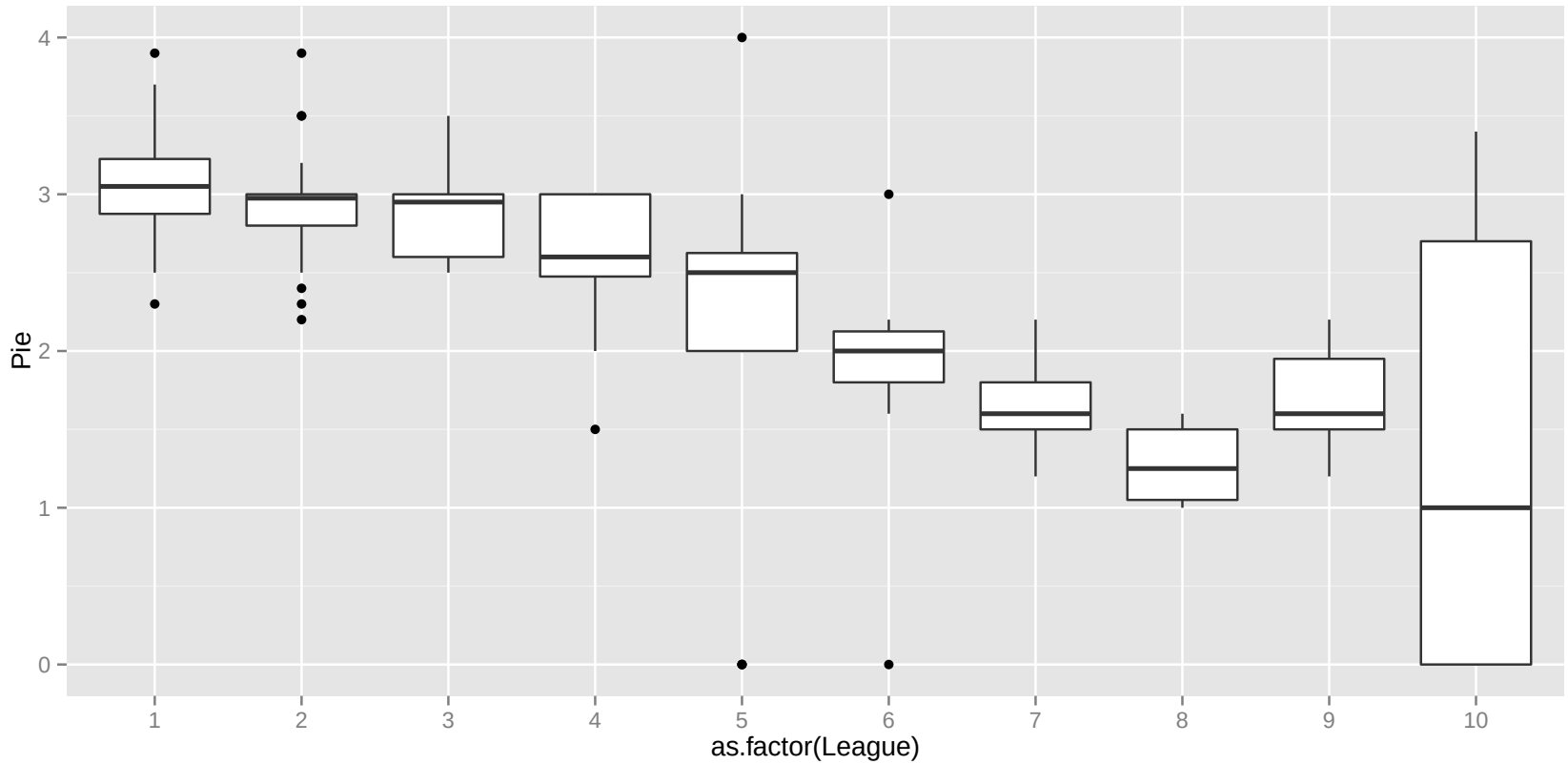
Jittering

```
ggplot(prices[sub, ], aes(x = Pie,
    y = Tea, color = factor(League))) +
    geom_jitter()
```



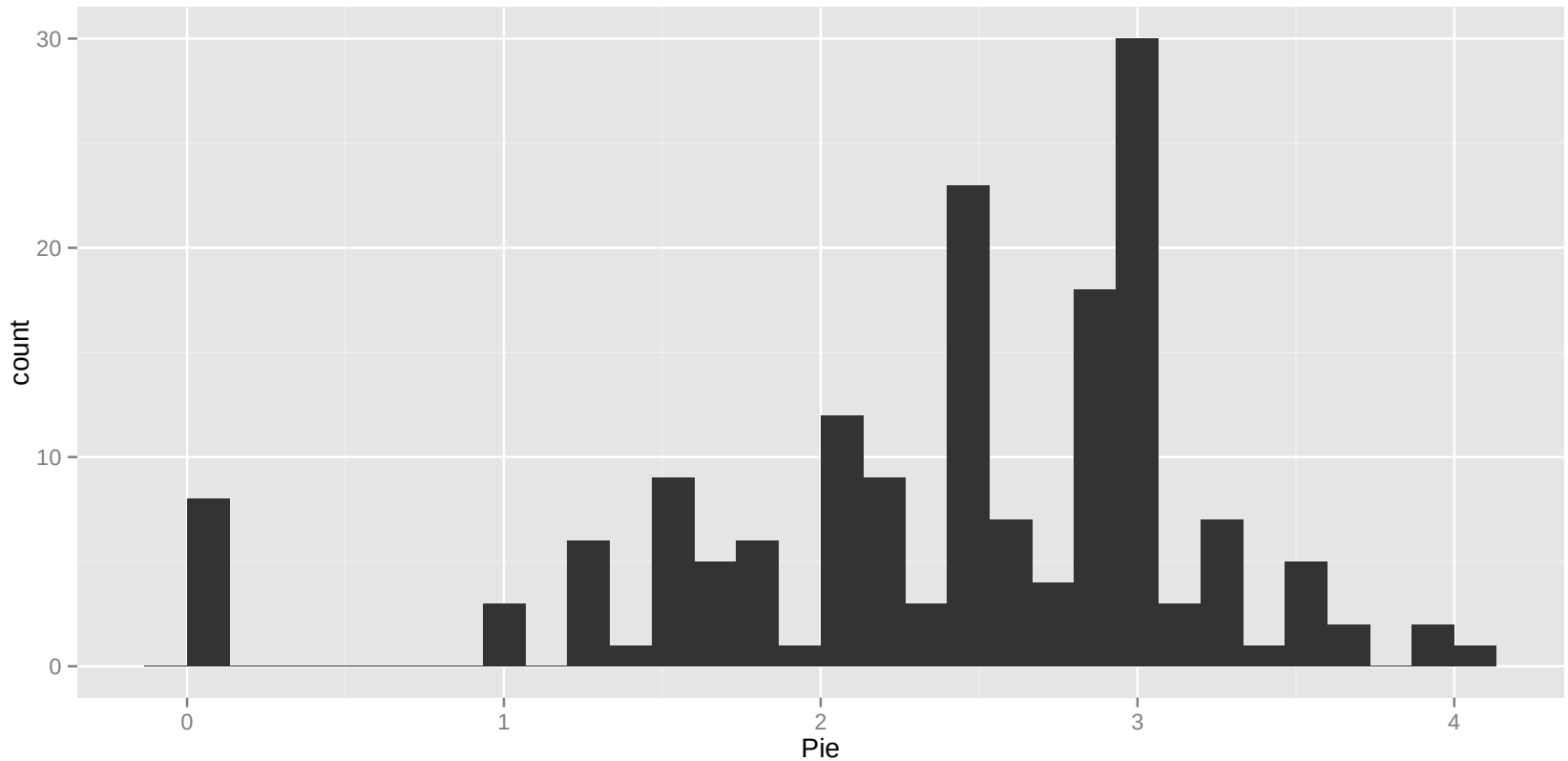
Boxplots

```
ggplot(prices, aes(x = as.factor(League),
    y = Pie)) + geom_boxplot()
```



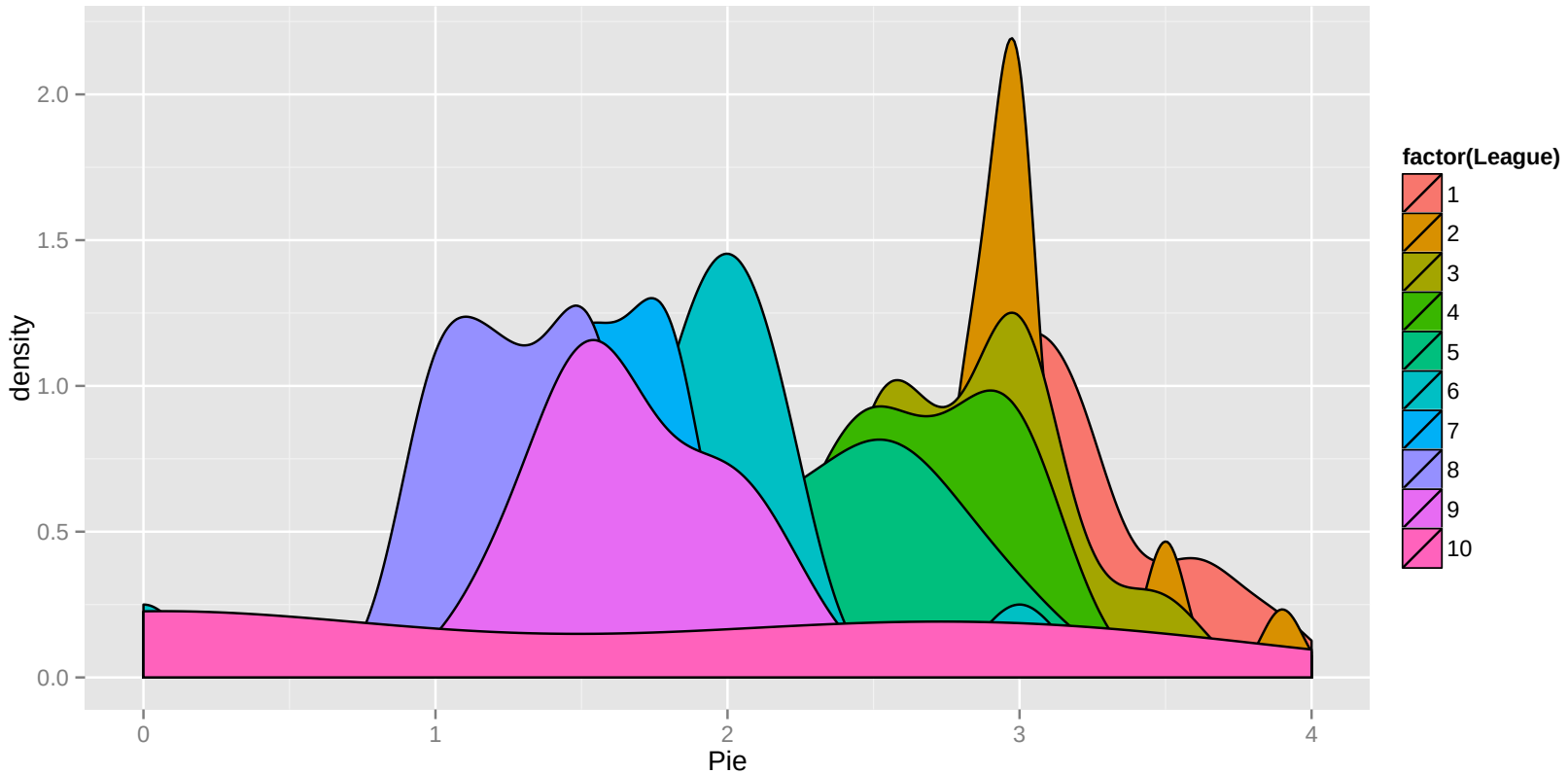
Histogram

```
ggplot(prices, aes(x = Pie)) +
    geom_histogram()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x'
to adjust this.
```



Density

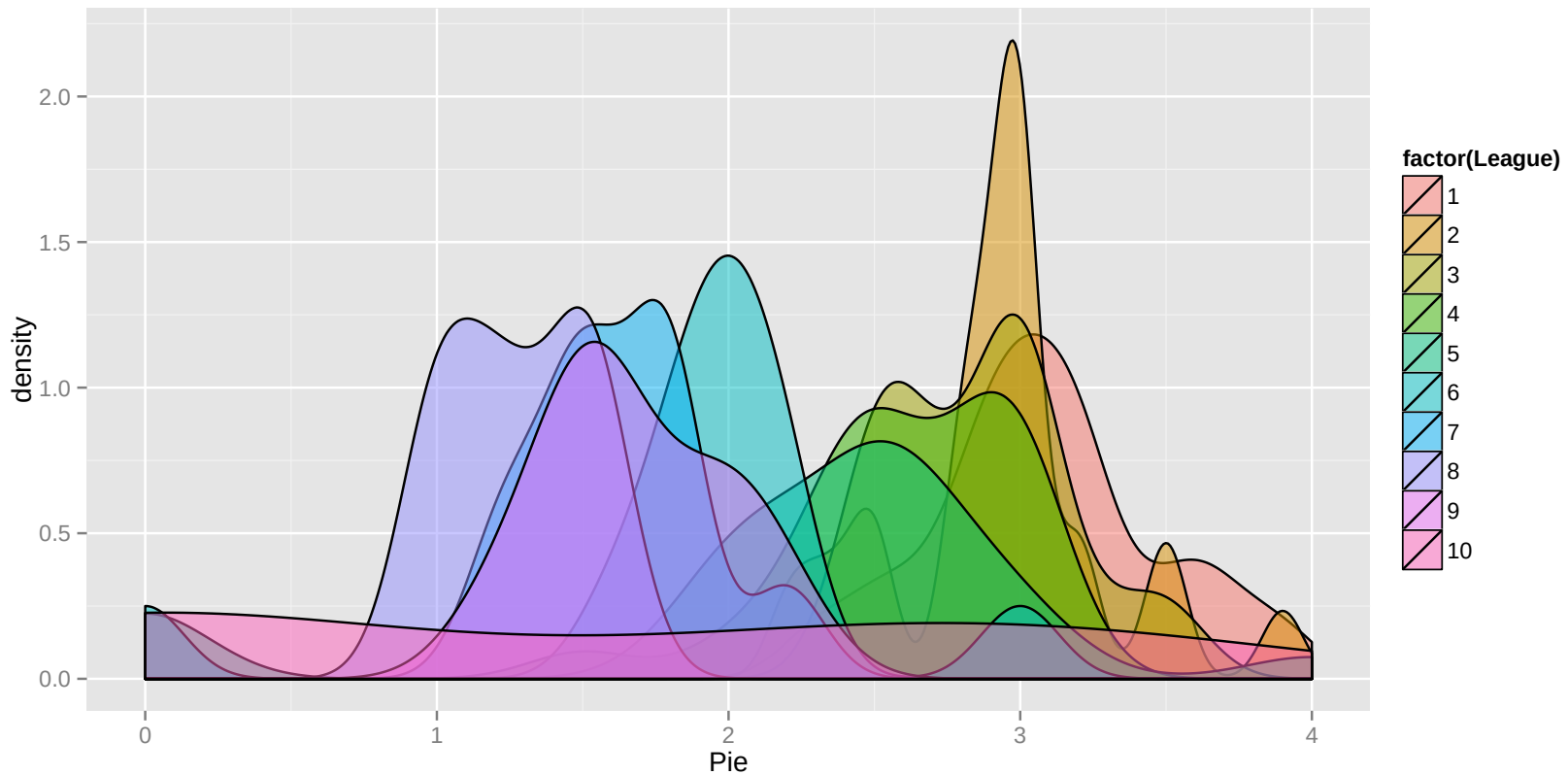
```
ggplot(prices, aes(x = Pie, fill = factor(League))) +
    geom_density()
```



Density

Can alter the transparency

```
ggplot(prices, aes(x = Pie, fill = factor(League))) +
    geom_density(alpha = 0.5)
```



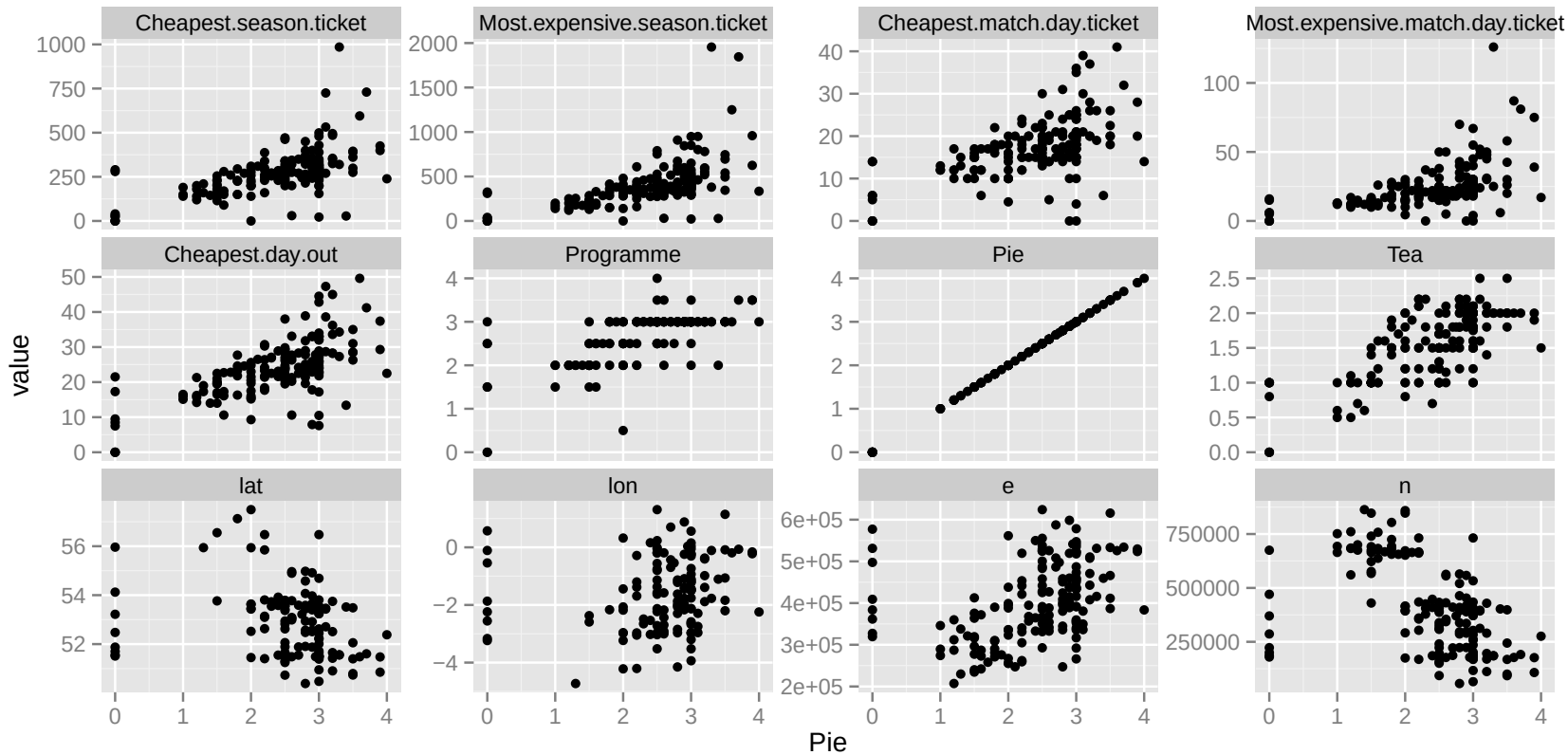
The shape of the data

ggplot prefers data to be in long format. We can 'melt' the data using the reshape library

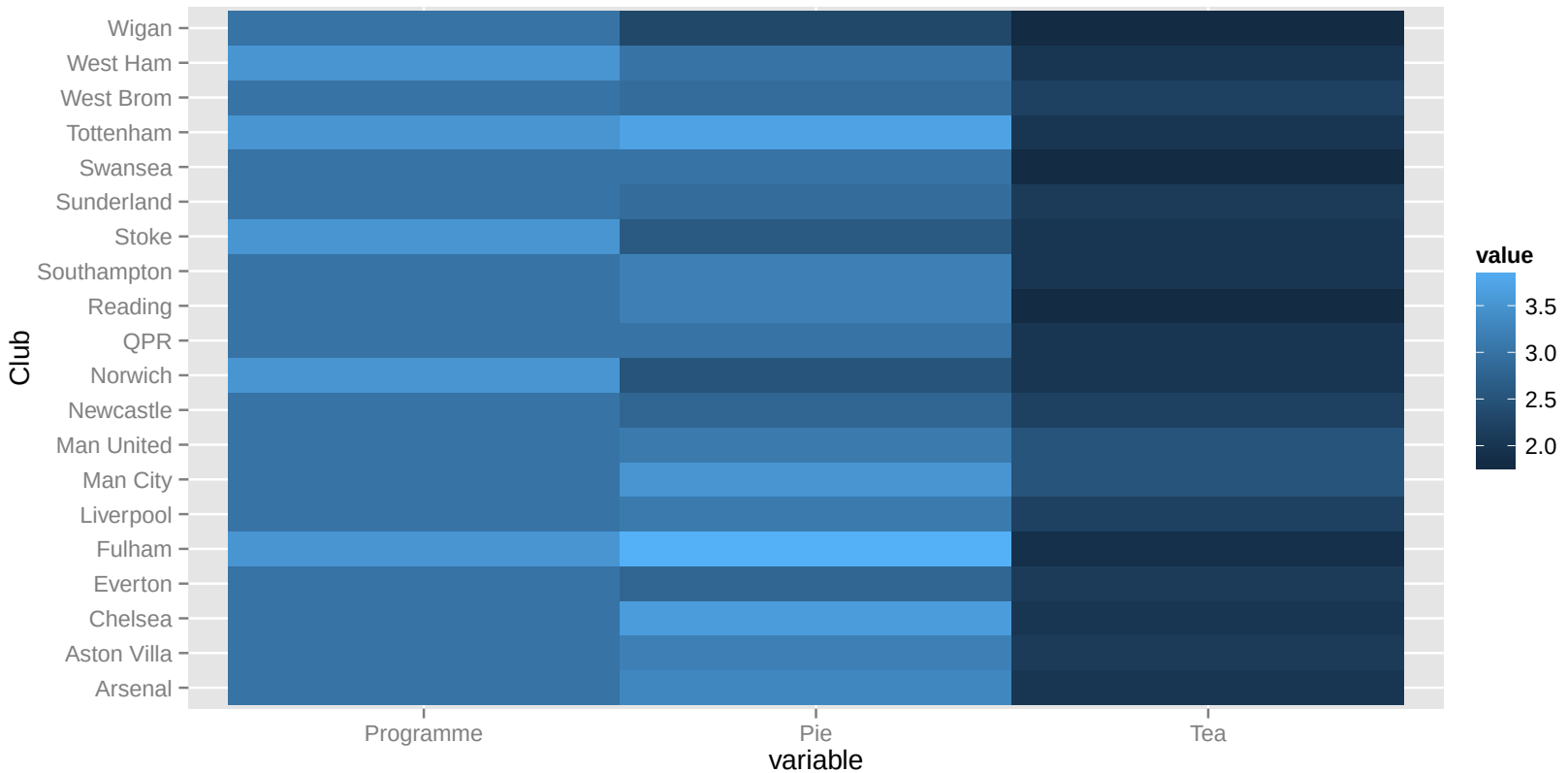
```
library(reshape)

mPrices <- melt(prices[, -2])

## Using Club as id variables

mPrices <- data.frame(mPrices,
    Pie = prices$Pie[match(mPrices[,
        1], prices[, 1])])
head(mPrices, 2)

##          Club
## 1     Arsenal
## 2 Aston Villa
##                  variable
## 1 Cheapest.season.ticket
## 2 Cheapest.season.ticket
##   value Pie
## 1   985 3.3
## 2   325 3.2
```

Getting really fancy

```
ggplot(mPrices, aes(x = Pie, y = value)) +
    geom_point() + facet_wrap(~variable,
    scales = "free_y")
```



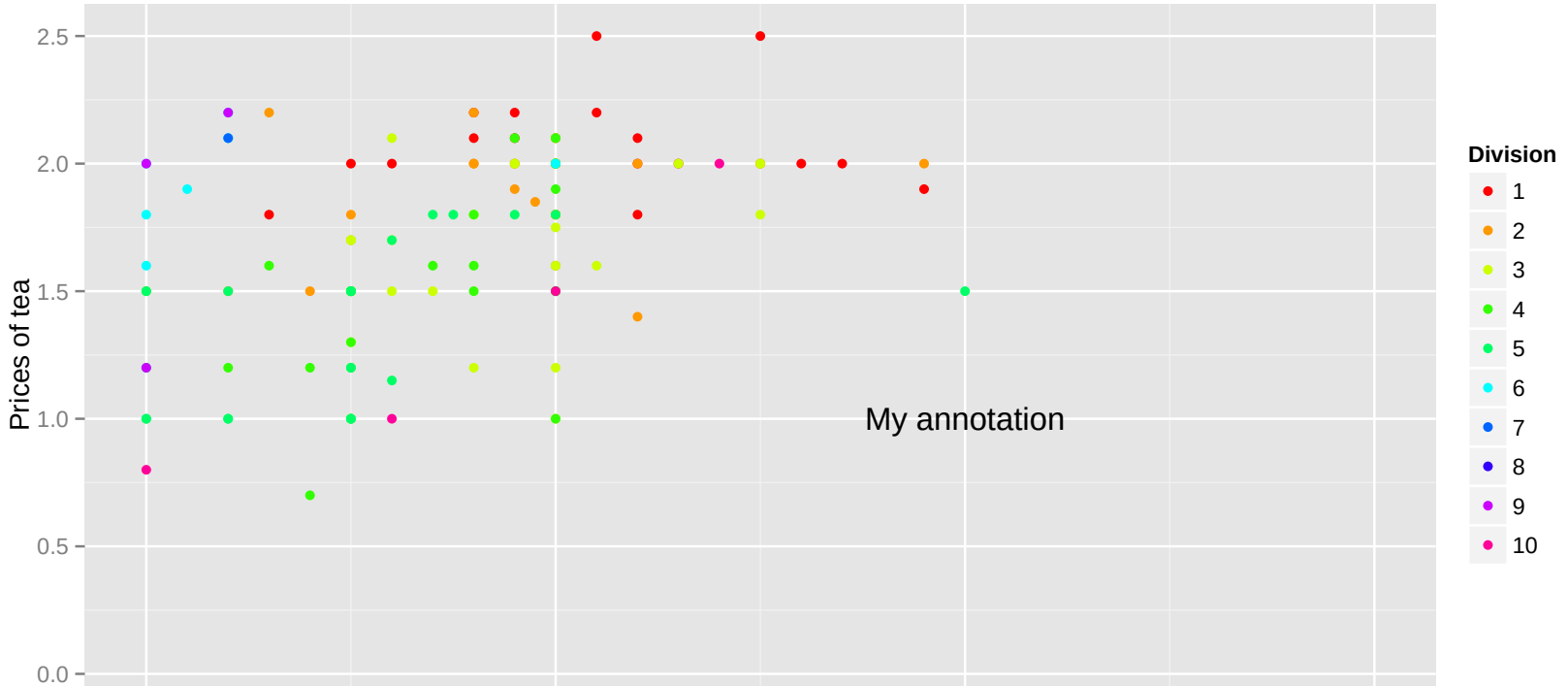
```
prem <- prices$League == 1
mPrices2 <- melt(prices[prem, c(1,
    8, 9, 10)])

## Using Club as id variables

ggplot(mPrices2, aes(x = variable,
    y = Club, fill = value)) +
    geom_tile()
```



Modification of plots

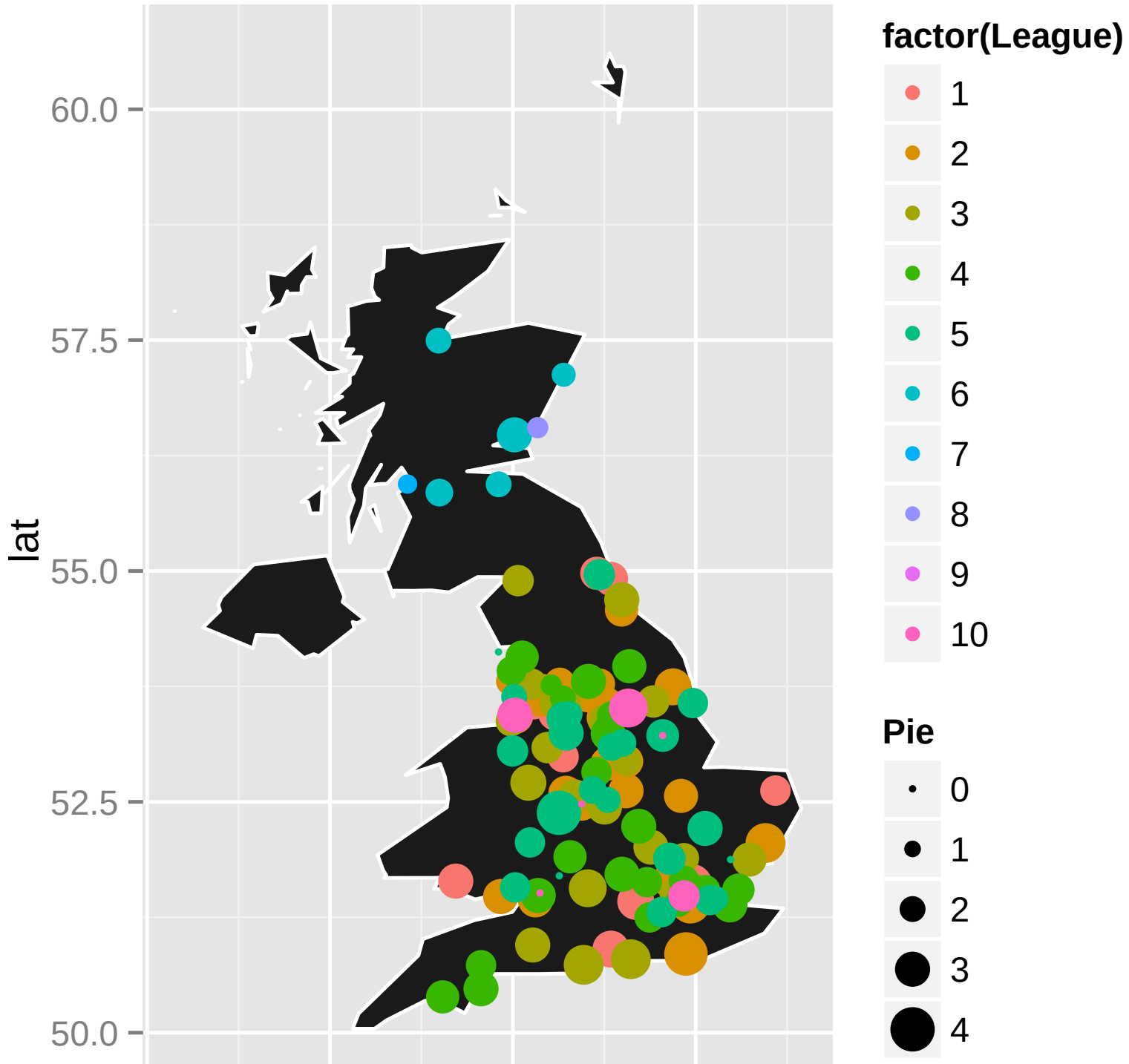
First command creates the plot, but doesn't display

```
p <- ggplot(prices, aes(x = Pie,
    y = Tea, color = as.factor(League))) +
    geom_point()

p + ylab("Prices of tea") + xlab("Price of Pies") +
    xlim(2, 5) + scale_color_manual(name = "Division",
    values = rainbow(10)) + annotate("text",
    4, 1, label = "My annotation")
```



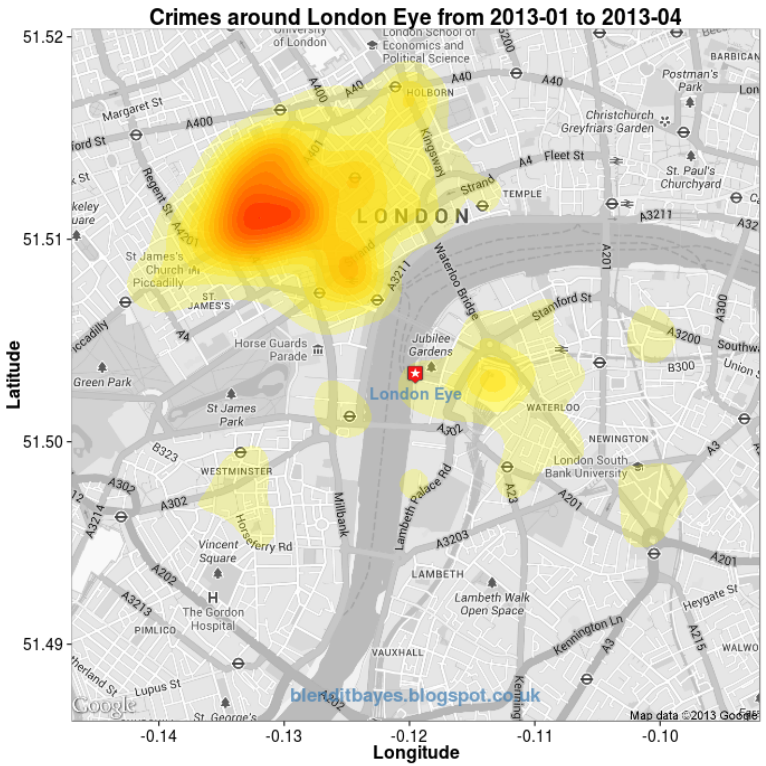
Finally...

```
uk <- map_data("world", region = "uk",
    xlim = c(-11, 3), ylim = c(51,
        60))
p <- ggplot()
p <- p + geom_polygon(data = uk,
    aes(x = long, y = lat, group = group),
    colour = "white", fill = "grey10")
```


lat
60.0
57.5
55.0
52.5
50.0
factor(League)
1
2
3
4
5
6
7
8
9
10
Pie
0
1
2
3
4

ggmap

Examples from the web



ggbio

http://www.tengfei.name/ggbio/
Bioconductor package for genomic visualisation based on ggplot2

- Manhatten plots
- Coverage plots
- Gene-models
- circos
- +many more