Data visualization

Mark Andrews

July 9, 2017

Introduction

R provides us with many tools for doing plotting and visualization. The dominant modern way is to use a program called `ggplot`, which is essentially a mini-language for data visualization.

```
library(readr)
library(ggplot2)
```

We'll read in our data

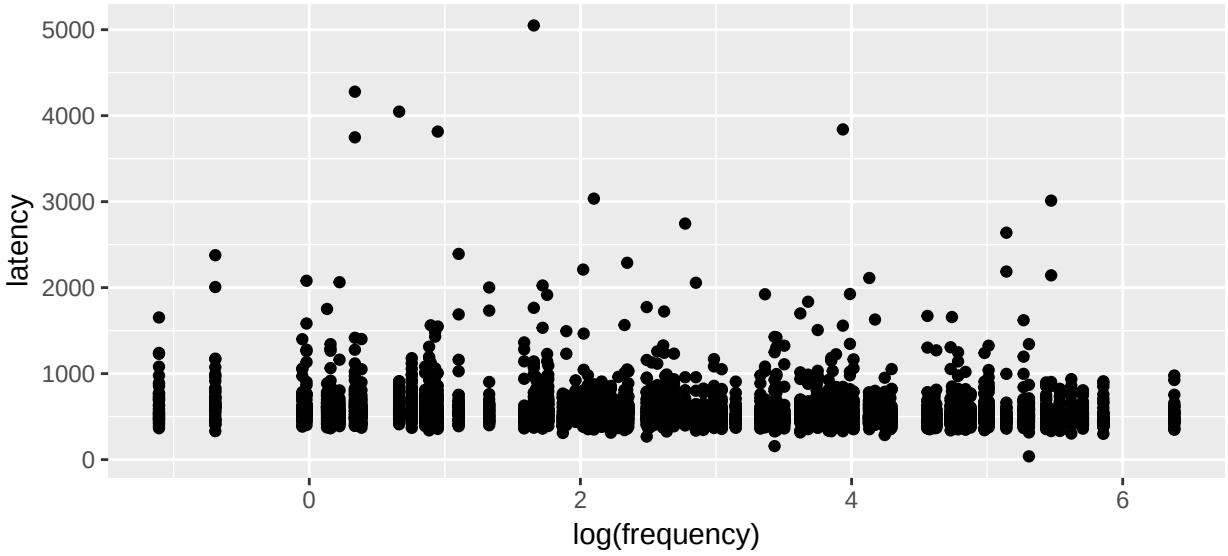
```
Df <- read_csv('../data/LexicalDecision.csv')
```

```
## Parsed with column specification:
## cols(
##   subject = col_integer(),
##   item = col_character(),
##   accuracy = col_integer(),
##   latency = col_integer(),
##   valence = col_double(),
##   length = col_integer(),
##   frequency = col_double()
## )
```

A simple scatter plot

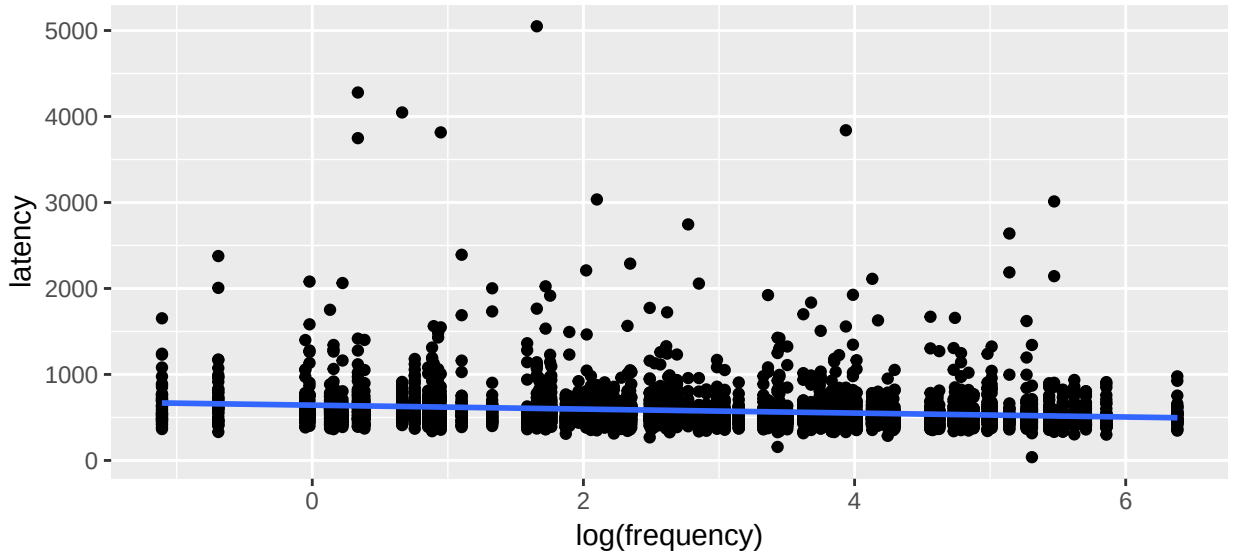
We use `ggplot` by first specifying the data frame to use and then which variables should be assigned to the x and y axes. Then we specify that we want points for the values.

```
ggplot(Df, aes(x=log(frequency), y=latency)) + geom_point()
```



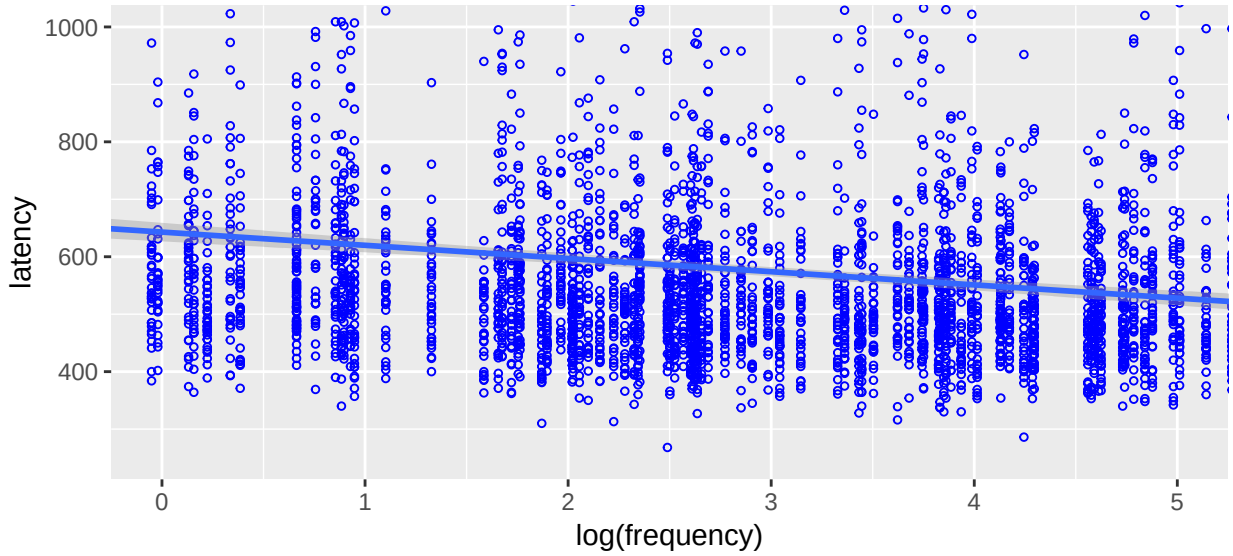
We can add another *layer* that is a line of best fit, plus standard error:

```
ggplot(Df, aes(x=log(frequency), y=latency)) +
  geom_point() +
  stat_smooth(method='lm')
```



We can play with the properties of all the features of our plot:

```
ggplot(Df, aes(x=log(frequency), y=latency)) +
  geom_point(size=1, shape=1, colour='blue') +
  stat_smooth(method='lm') +
  coord_cartesian(xlim = c(0, 5), ylim=c(250, 1000))
```

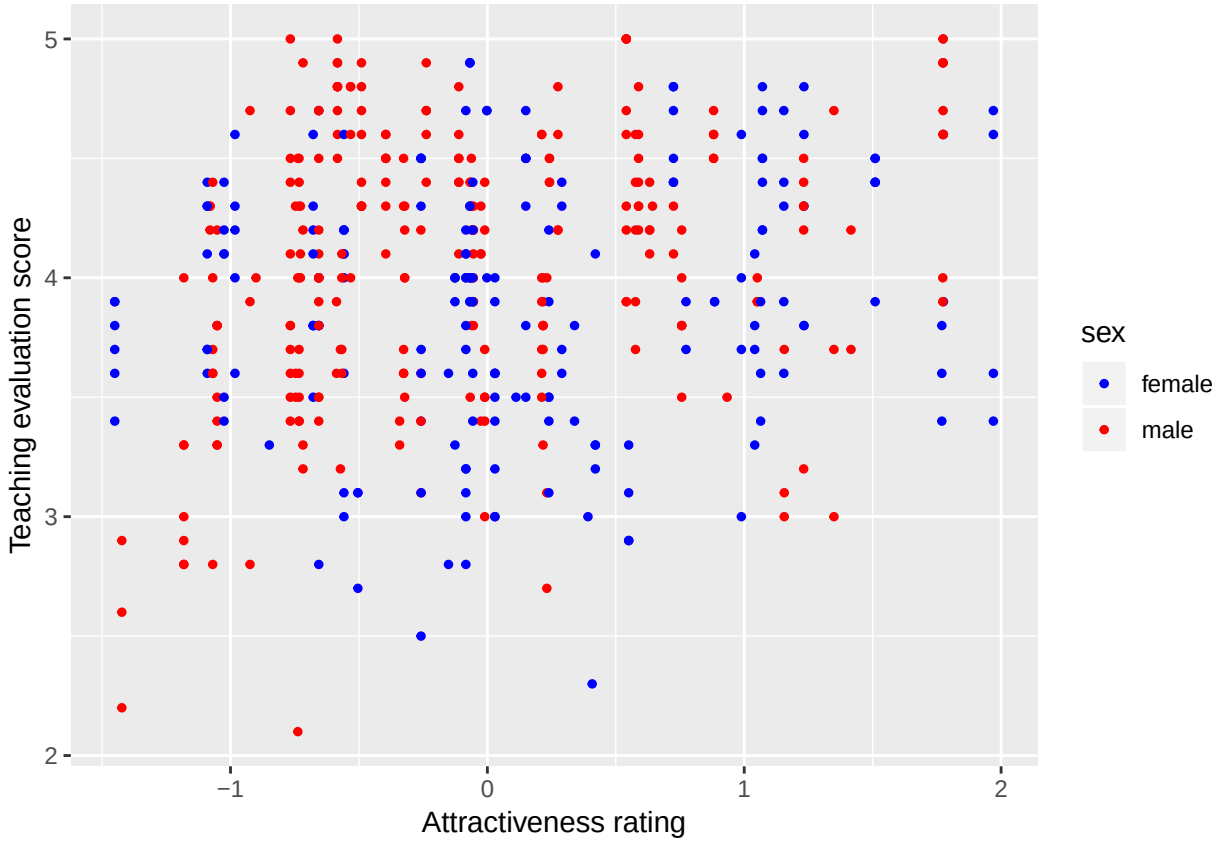


Colour and shape codes

One of the great things about `ggplot` is how it allows us to colour code our data. For example, here we make a scatter plot and color code the points that belong to males and females. (When using alternative plotting methods to `ggplot`, doing things like this become very cumbersome):

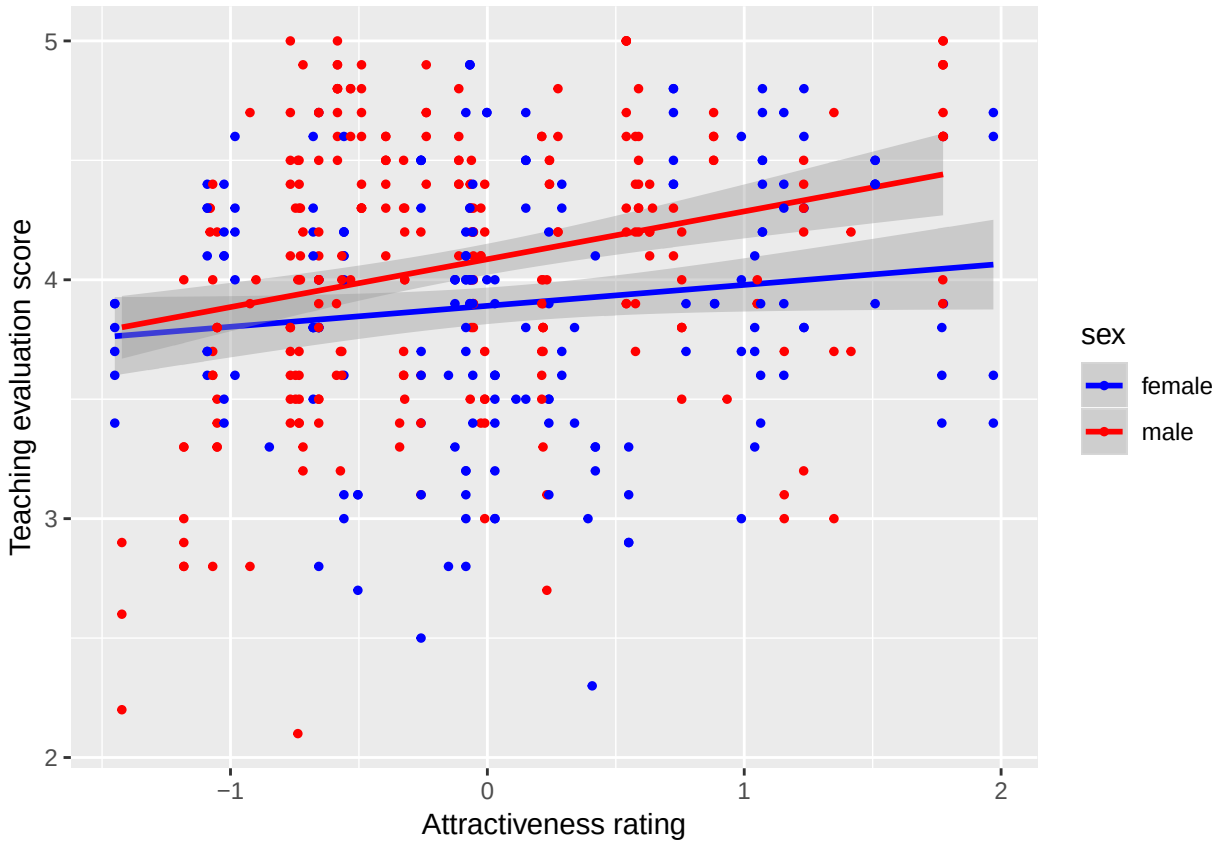
```
# Load a new data set
load('../data/beautyeval.Rda')

# Make scatterplot, point colour by male/female
ggplot(beautydata, aes(x=beauty, y=eval, colour=sex)) +
  geom_point(size=1) +
  scale_color_manual(values=c('blue', 'red')) +
  scale_y_continuous(name='Teaching evaluation score') +
  scale_x_continuous((name='Attractiveness rating'))
```



As before, we can superimpose lines of best fit, etc.

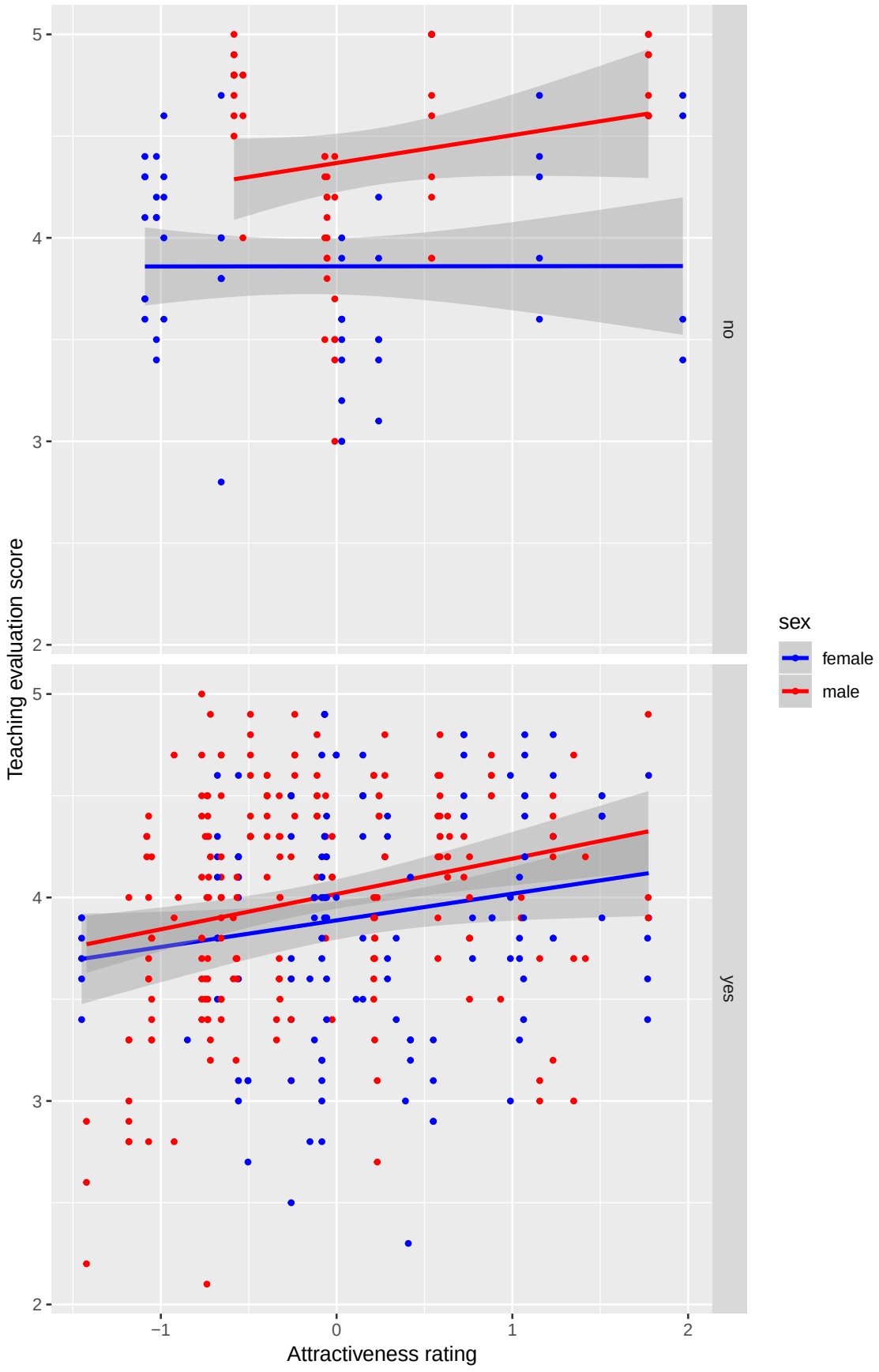
```
# Scatterplot, with lines of best fit and errors
ggplot(beautydata, aes(x=beauty, y=eval, colour=sex)) +
  stat_smooth(method='lm') +
  geom_point(size=1) +
  scale_color_manual(values=c('blue', 'red')) +
  scale_y_continuous(name='Teaching evaluation score') +
  scale_x_continuous((name='Attractiveness rating'))
```



Facets

Facets allow us to create multiple plots in the same way, with each one showing some subset of the data.

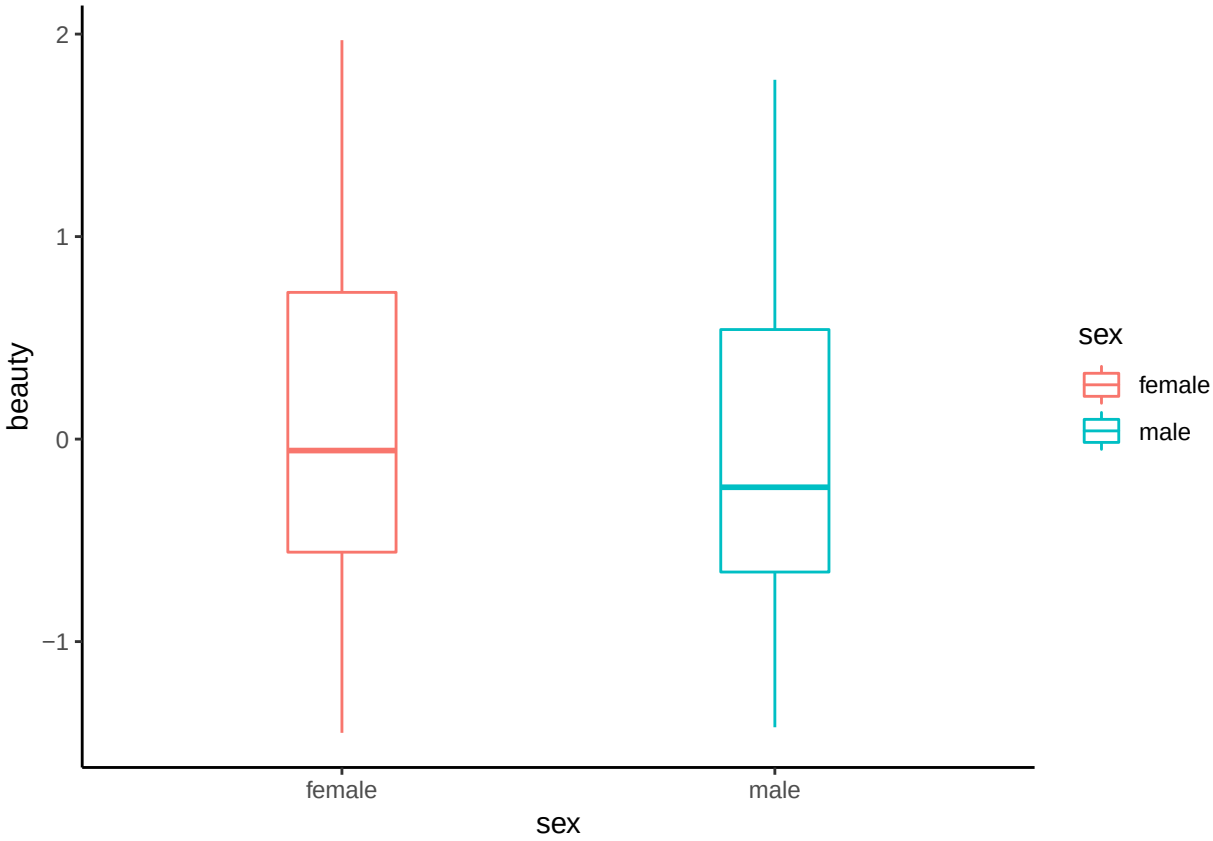
```
# Scatterplot, with lines of best fit and errors
# one per tenure group
ggplot(beautydata, aes(x=beauty, y=eval, colour=sex)) +
  stat_smooth(method='lm') +
  geom_point(size=1) +
  scale_color_manual(values=c('blue', 'red')) +
  scale_y_continuous(name='Teaching evaluation score') +
  scale_x_continuous((name='Attractiveness rating')) +
  facet_grid(tenure ~ .)
```


Teaching evaluation score
Attractiveness rating
5
4
3
2
−1
0
1
2
no
yes
sex
female
male

Plotting distributions

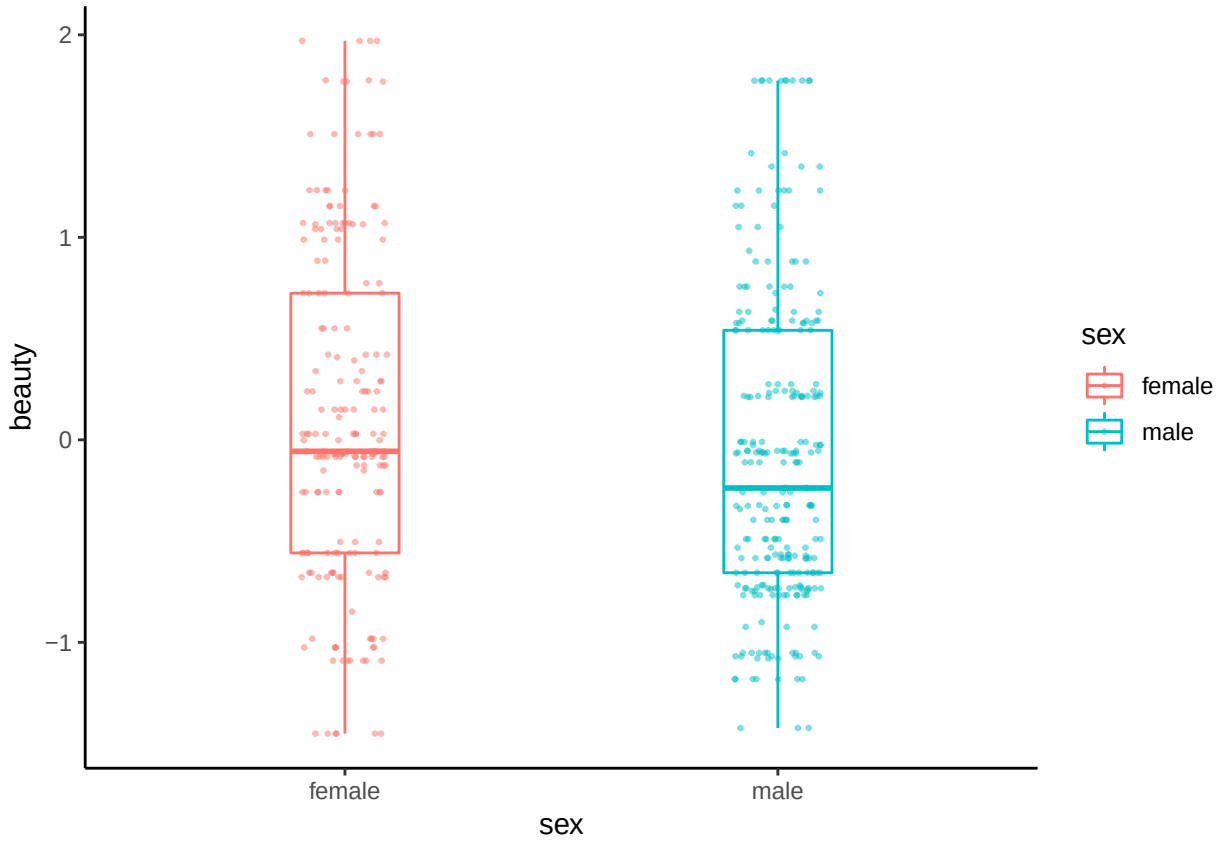
So far, we've focused on scatter and line plots. These are, of course, very common. But equally common are plots showing distributions of univariate data. Of these, the Tukey boxplot is an oldie but (in my opinion) still a goodie.

```
ggplot(beautydata,
       aes(x = sex, y = beauty, col = sex)) +
  geom_boxplot(width=0.25) +
  theme_classic()
```



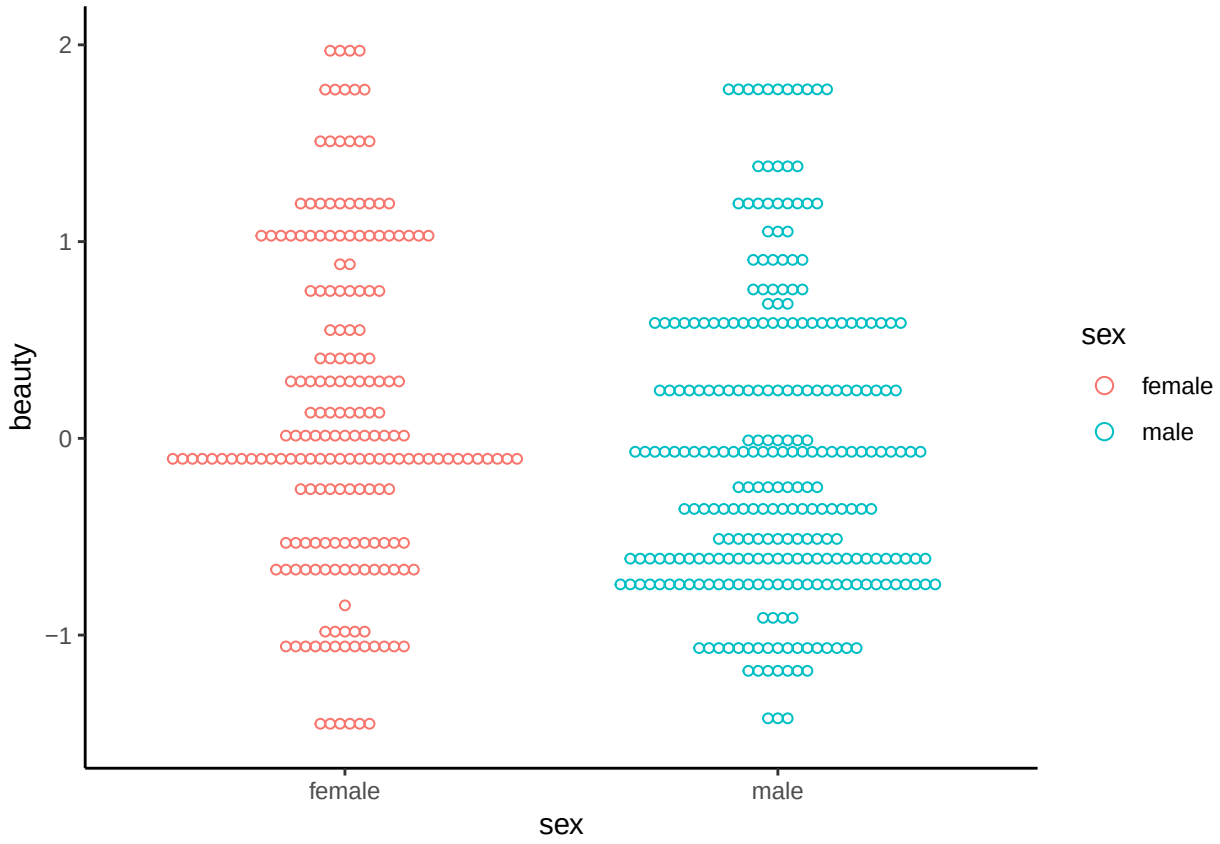
It is generally a good idea to show as much of the raw data as possible, so we can do that, while still keeping the boxplot, using a *jittered* dot display too.

```
ggplot(beautydata,
       aes(x = sex, y = beauty, col = sex)) +
  geom_boxplot(width=0.25) +
  geom_jitter(width = 0.1, size=0.5, alpha=0.5) +
  theme_classic()
```



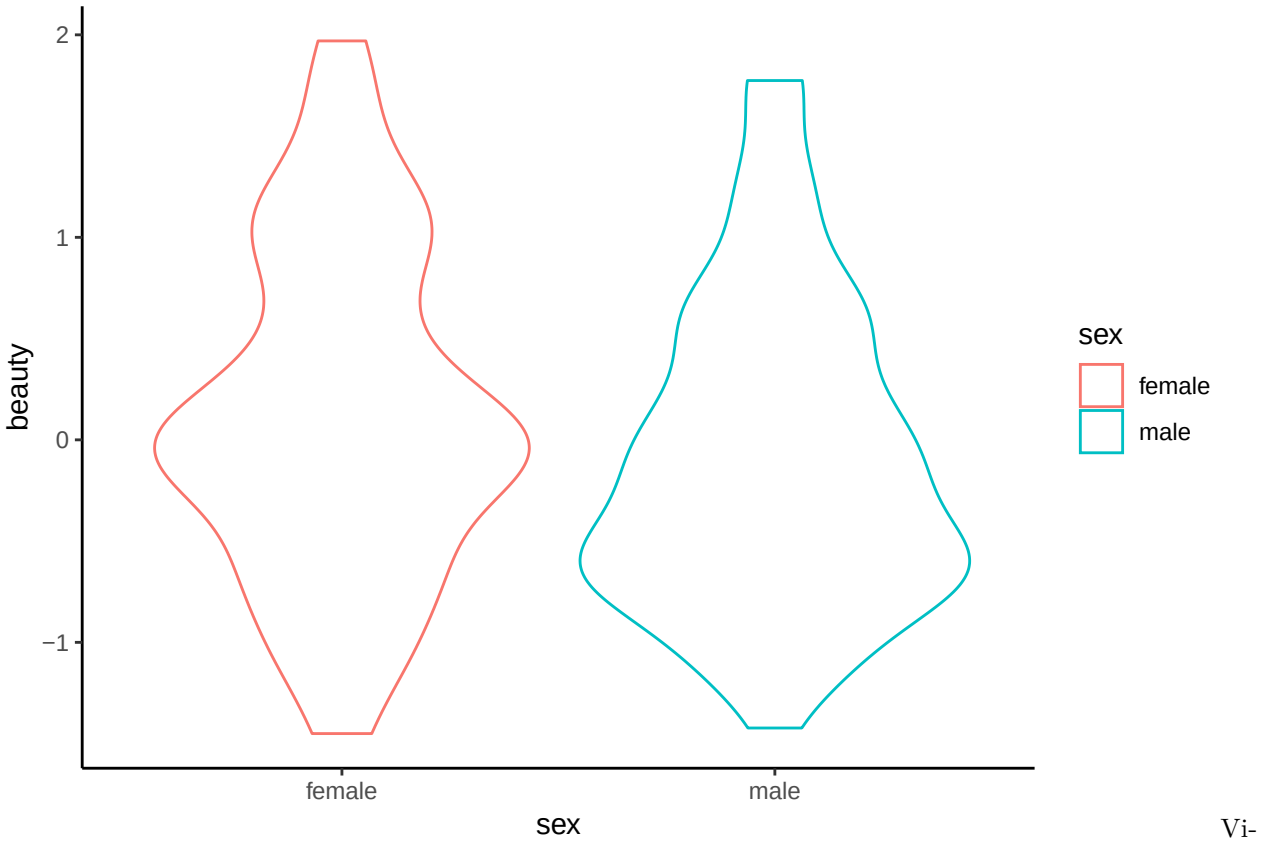
Another option that might be sometimes useful is a dotplot.

```
ggplot(beautydata,
       aes(x = sex, y = beauty, col = sex)) +
  geom_dotplot(binaxis = 'y', width=0.25, fill=NA, stackdir = 'center', dotsize = 0.5, binwidth=0.1) +
  theme_classic()
```



Yet another is a so-called *violon* plot:

```
ggplot(beautydata,
       aes(x = sex, y = beauty, col = sex)) +
  geom_violin() +
  theme_classic()
```



Violin plots are a relatively recent invention, but I think violon plots are a bit overrated. Personally, I usually prefer a good old fashioned boxplot with jittered points.