Advanced stat functions

Jumping Rivers

Perhaps the easiest `stat_*` to consider is the `stat_summary()` function. This function summarises `y` values at every unique `x` value. This is quite handy, for example, when adding single points that summarise the data or adding error bars.

A simple plot to create, is the mean alcohol consumption per actor (figure 1)

```
library("ggplot2")
data(bond, package = "jrGgplot2")
ggplot(bond, aes(Actor, Alcohol_Units)) + stat_summary(geom = "point",
    fun.y = mean)
```

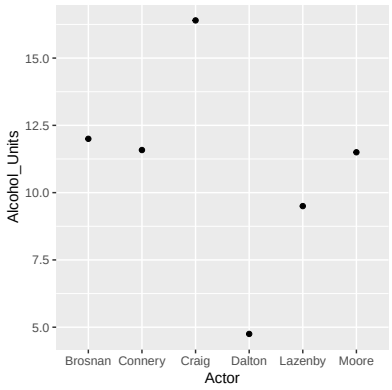

Figure 1: Average number of units consumed per actor.

In the above piece of code we calculate the mean number of alcohol units consumed by each Actor. These x-y values are passed to the `point` geom. We can use any function for `fun.y` provided it takes in a vector and returns a single point. For example, we could calculate the range of values, as in figure 2:

```
ggplot(bond, aes(Actor, Alcohol_Units)) +
    stat_summary(geom = "point",
                 fun.y =  function(i) max(i) - min(i))
```

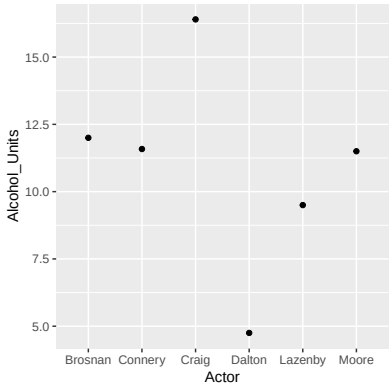

Figure 2: Plot of the range for each actor.

Or we could work out confidence intervals for the mean number of Units consumed (figure 13):

```
## Standard error function
std_err = function(i)
    dt(0.975, length(i) - 1) * sd(i) / sqrt(length(i))

ggplot(bond, aes(x = Actor, y = Alcohol_Units)) +
    stat_summary(fun.ymin = function(i) mean(i) - std_err(i),
                 fun.ymax = function(i) mean(i) + std_err(i),
                 colour = "steelblue", geom = "errorbar",
                 width = 0.2, lwd = 2) +
    ylim(c(0, 20))
```

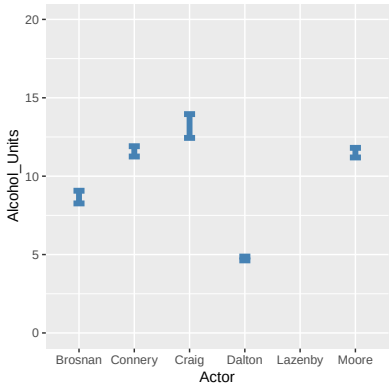

Figure 3: Confidence intervals for the mean number of units consumed by each actor.

To calculate the bounds, we work out the standard deviation (`sd(i)`), then number of movies per actor (`length(i)`) and the correct value from the t distributions, with $n - 1$ degrees of freedom.