Practical 6

Adele Valeria

November 10, 2019

Visualizing Data

1. Overplotting

1.1 Import data

```
library(ggplot2)
data("diamonds")
```

1.2 Scatterplot

The dots (points) are overlaid. We should resize the point size or change the transparency to make the plot look better.

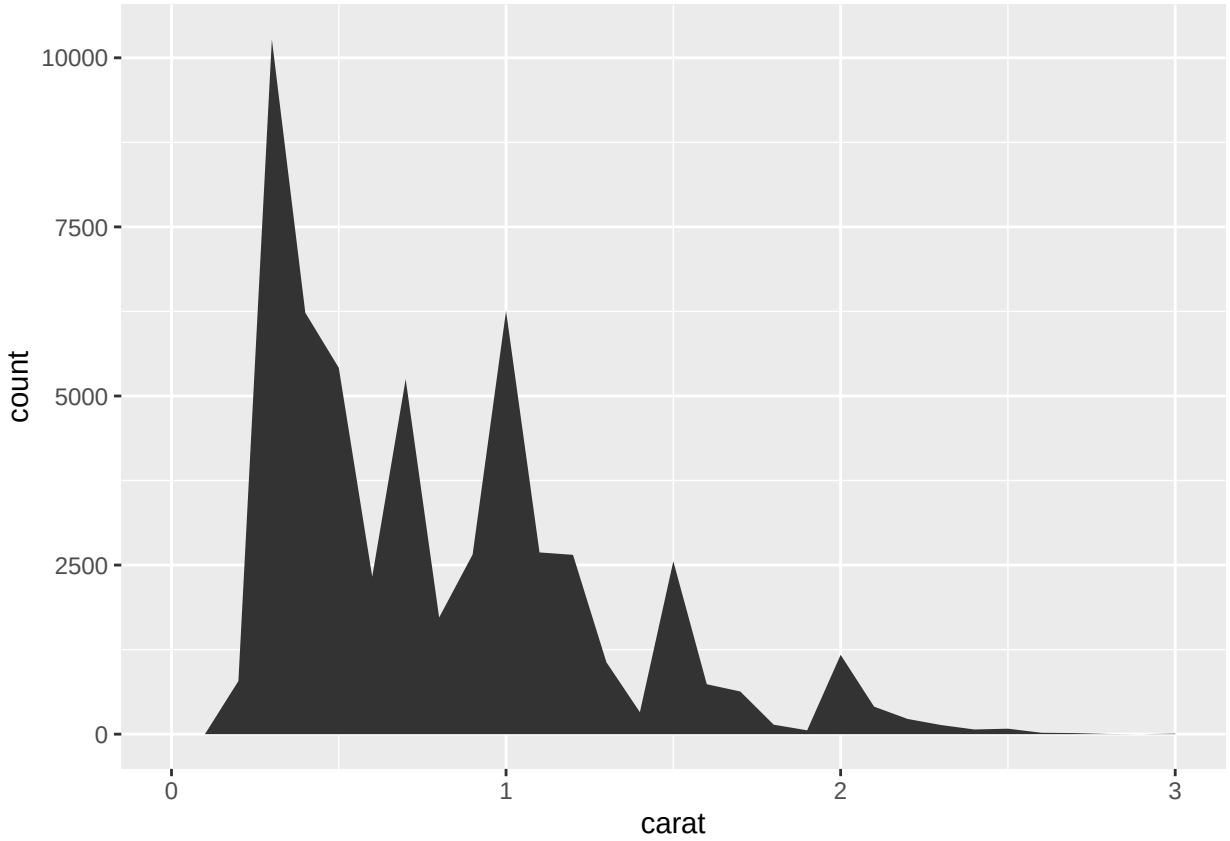
```
d <- ggplot(diamonds, aes(x= carat,y=price,colour=cut))
d + geom_point()
```

```
#Resize the point size to 0.1
d + geom_point(size=0.1)
```

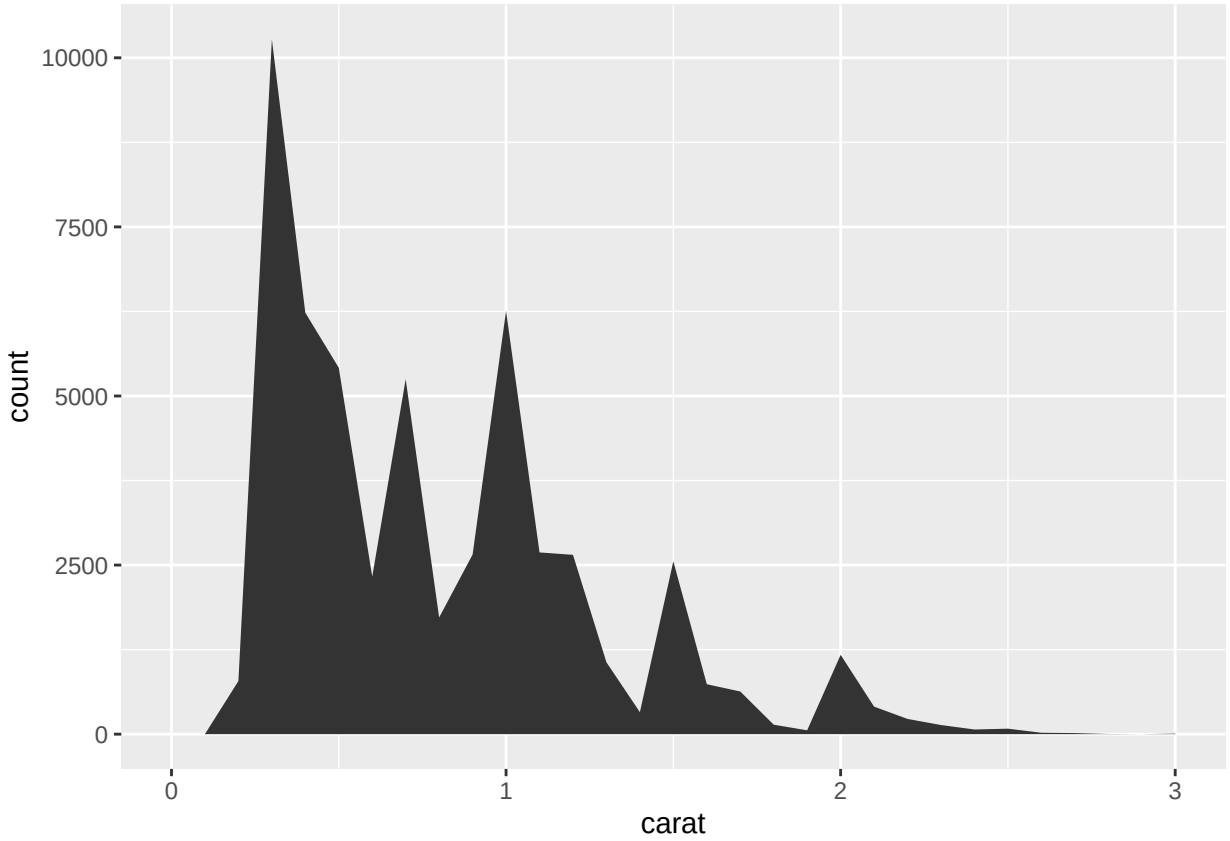
```
#Change the transparency to alpha 1/5
#Change the shape of the point by setting the shape argument to another number, such as 18
d + geom_point(alpha=1/5, shape = 18)
```

2. geom_XXX and stat_xxx

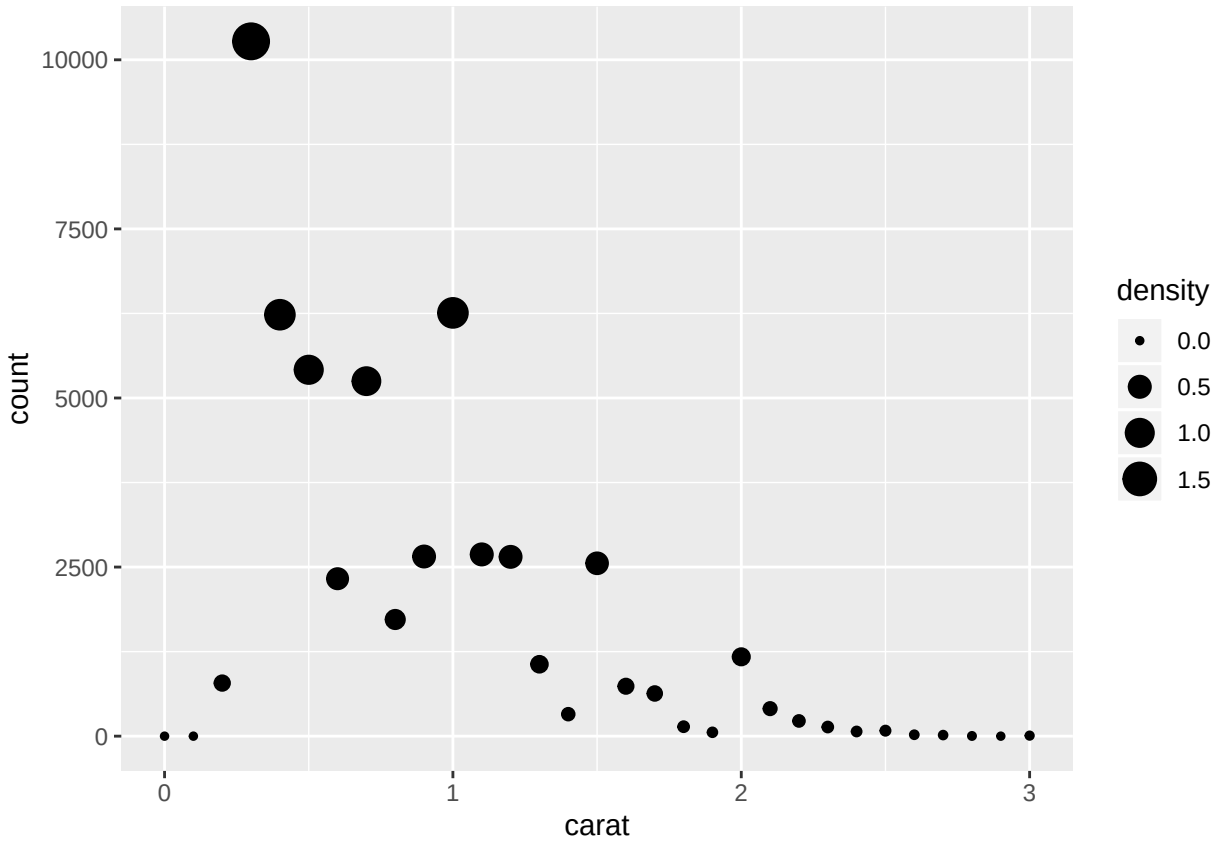
```
d <- ggplot(diamonds, aes(carat)) + xlim(0, 3)
d + stat_bin(aes(ymax = ..count..), binwidth = 0.1, geom = "area")
```



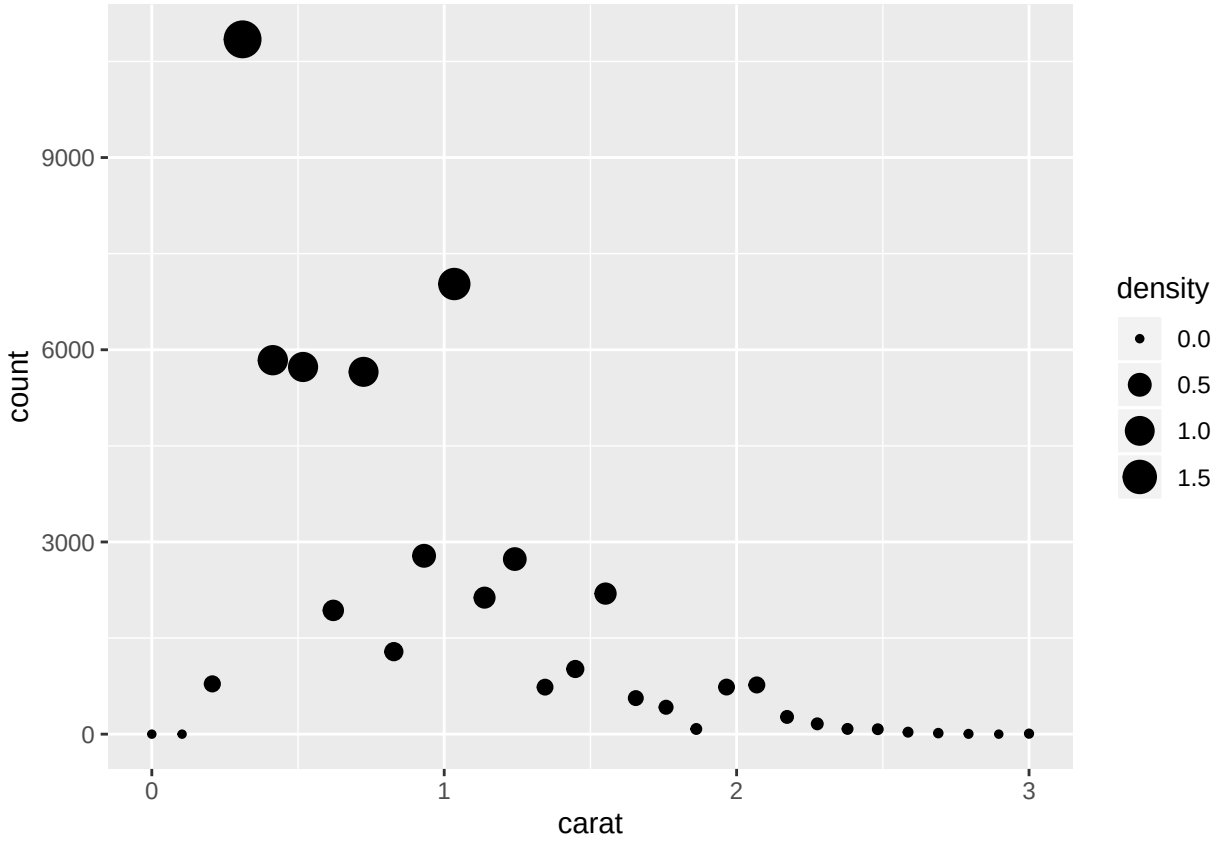

```
d + geom_area(aes(ymax=..count..), binwidth =0.1, stat = "bin")
```




```
d + stat_bin(aes(size = ..density..), binwidth = 0.1, geom = "point", position="identity" )
```

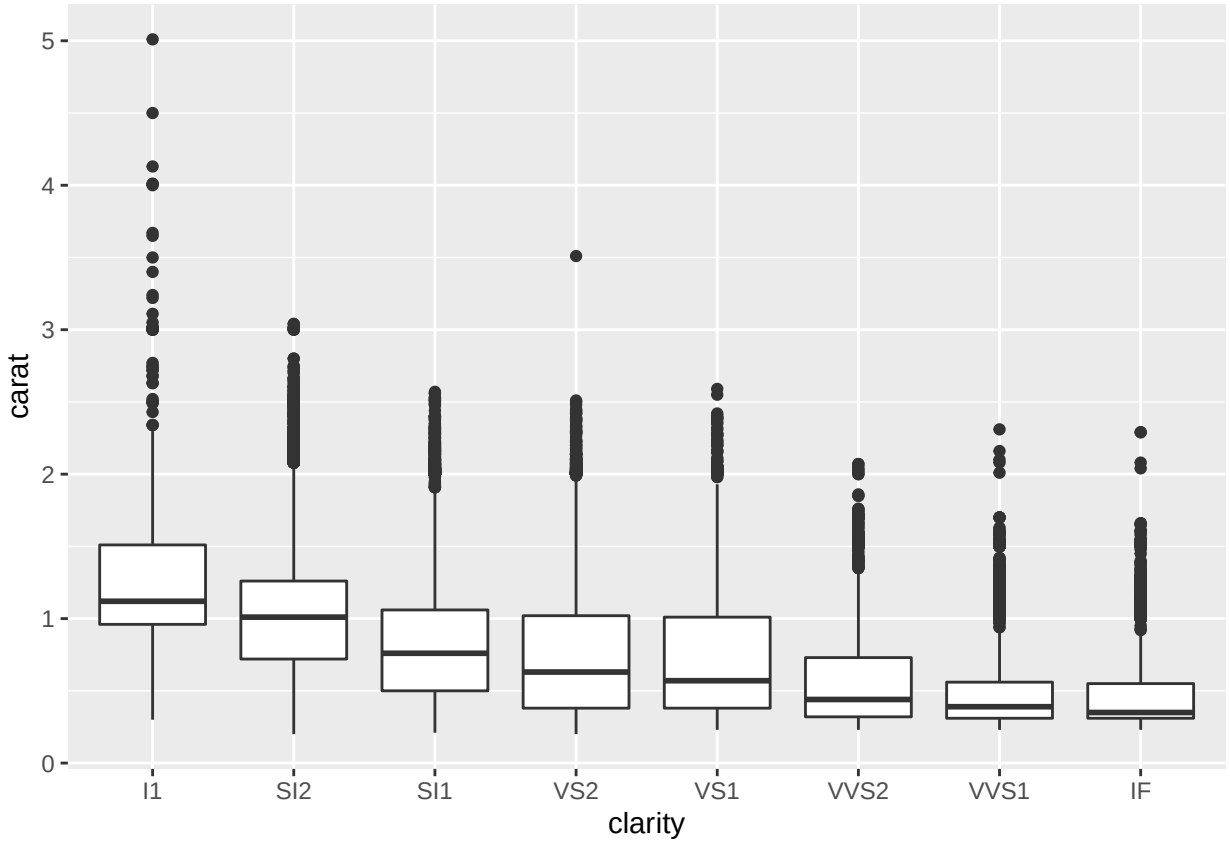



```
d+ geom_point(aes(size=..density..), position="identity", stat = "bin", binwith=0.1)
```

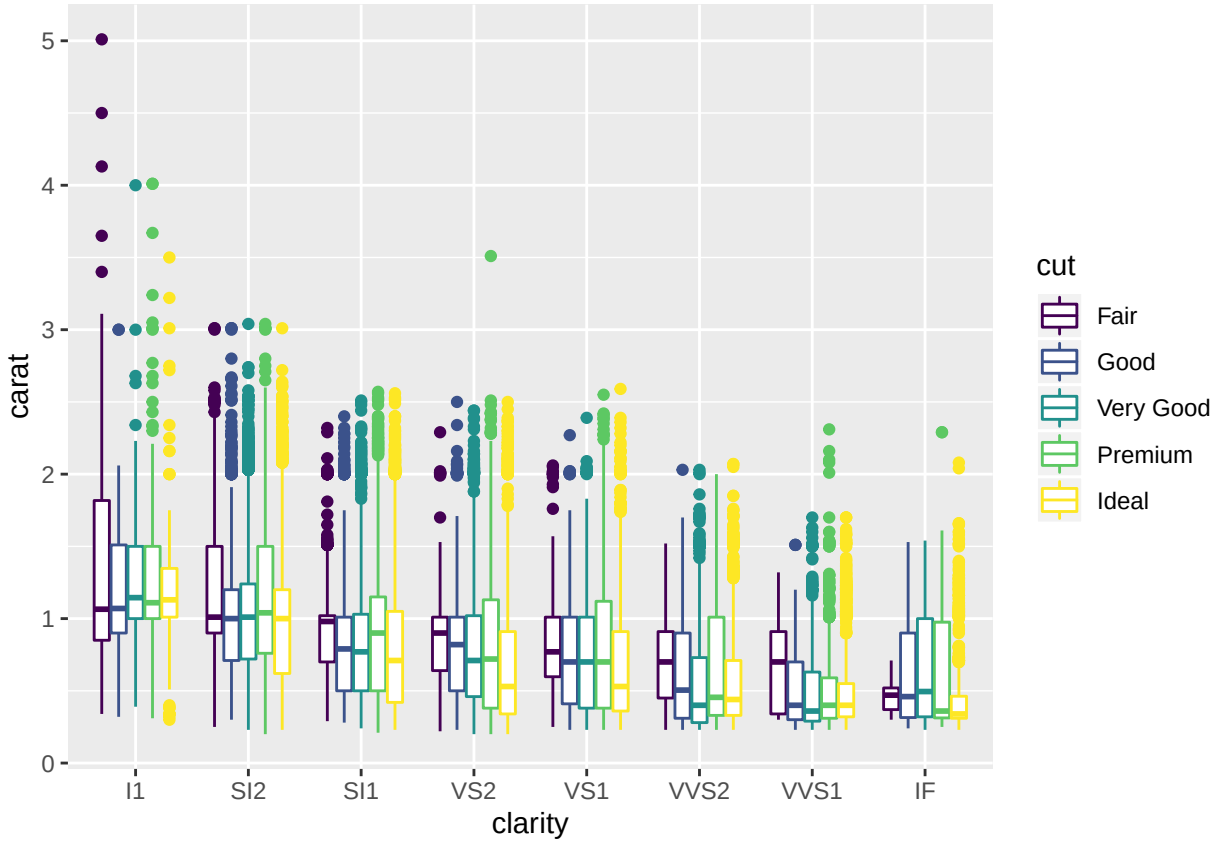


3. Build plots layer by layer

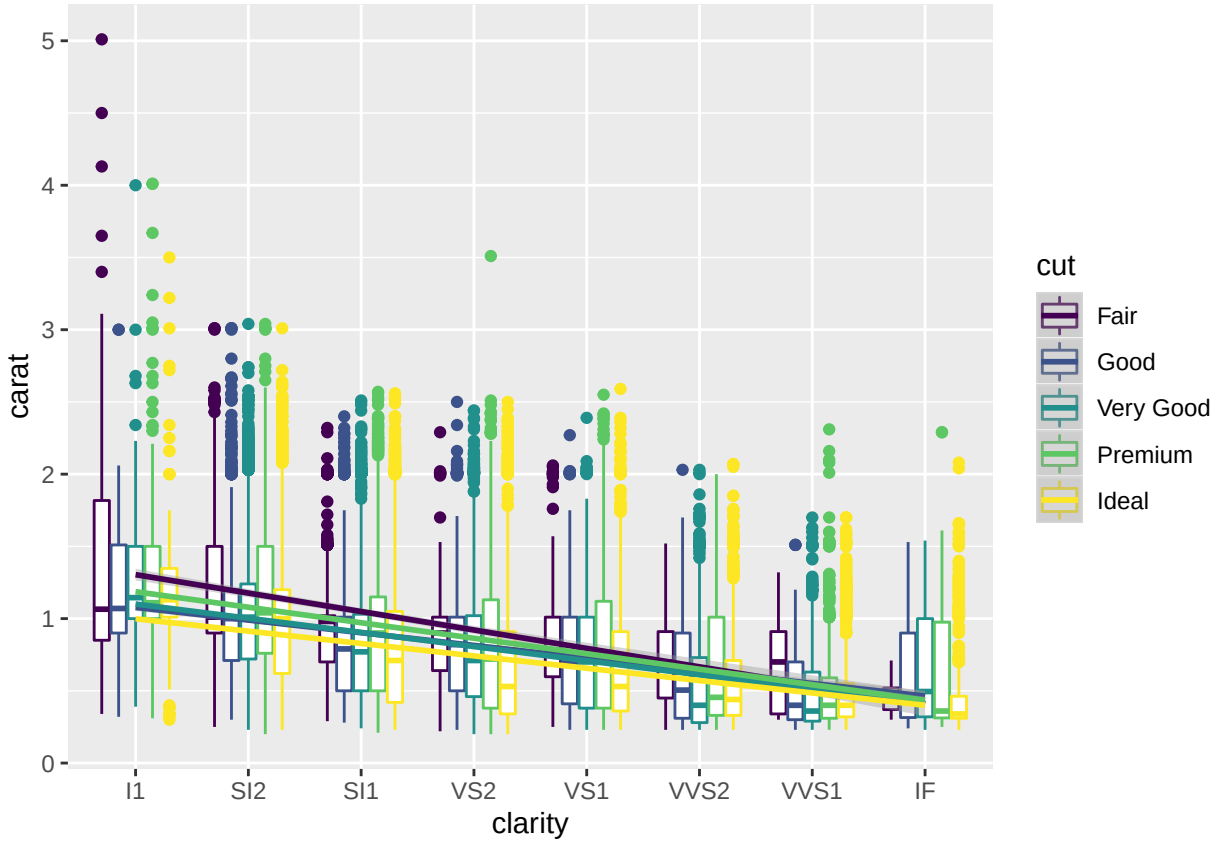
```
##layer by layer
d <- ggplot(diamonds, aes(x= clarity, y= carat))
d + geom_boxplot()
```



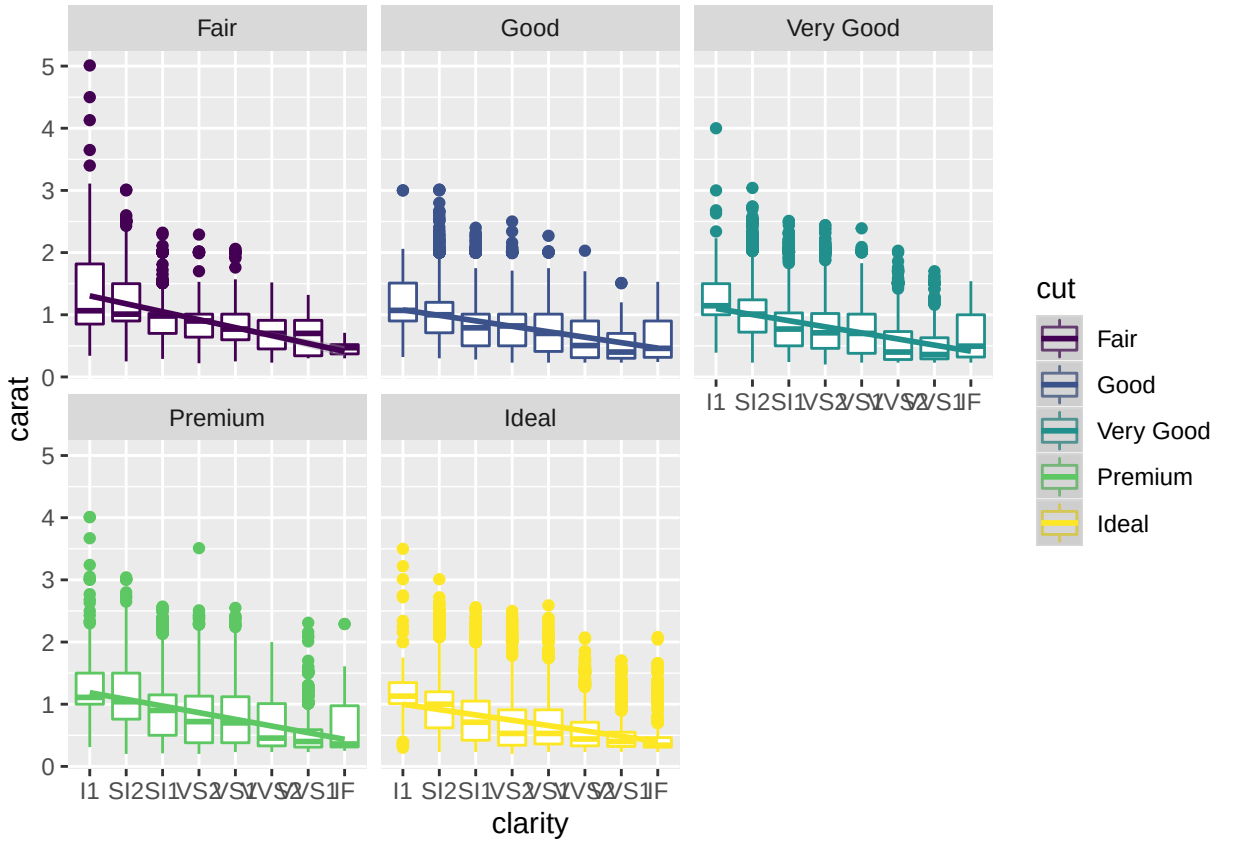

```
d + geom_boxplot(aes(colour=cut))
```



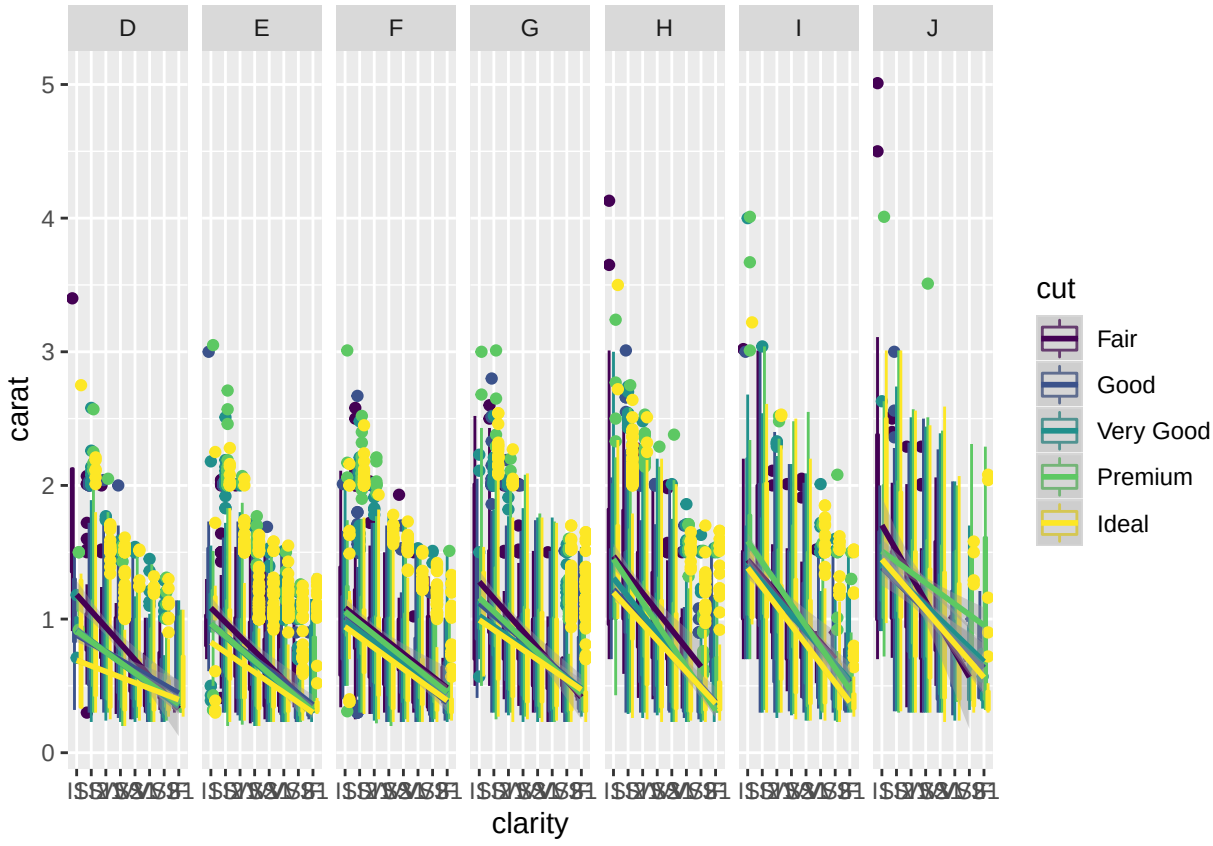

```
#geom_smooth, use method = lm
sm <- geom_smooth(aes(group=cut,colour=cut),method = "lm")
d1 <- d + geom_boxplot(aes(colour=cut)) + sm
d1
```



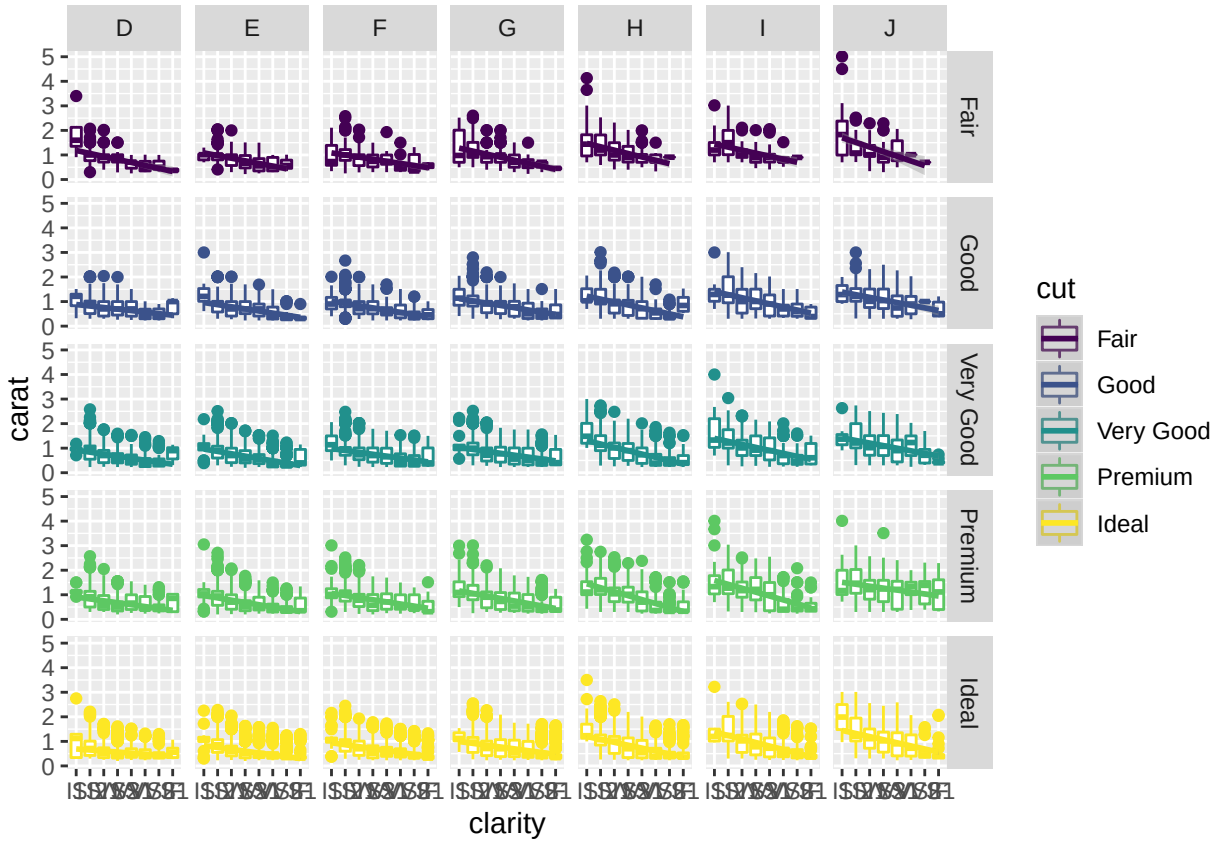

```
d1 + facet_wrap(~cut)
```



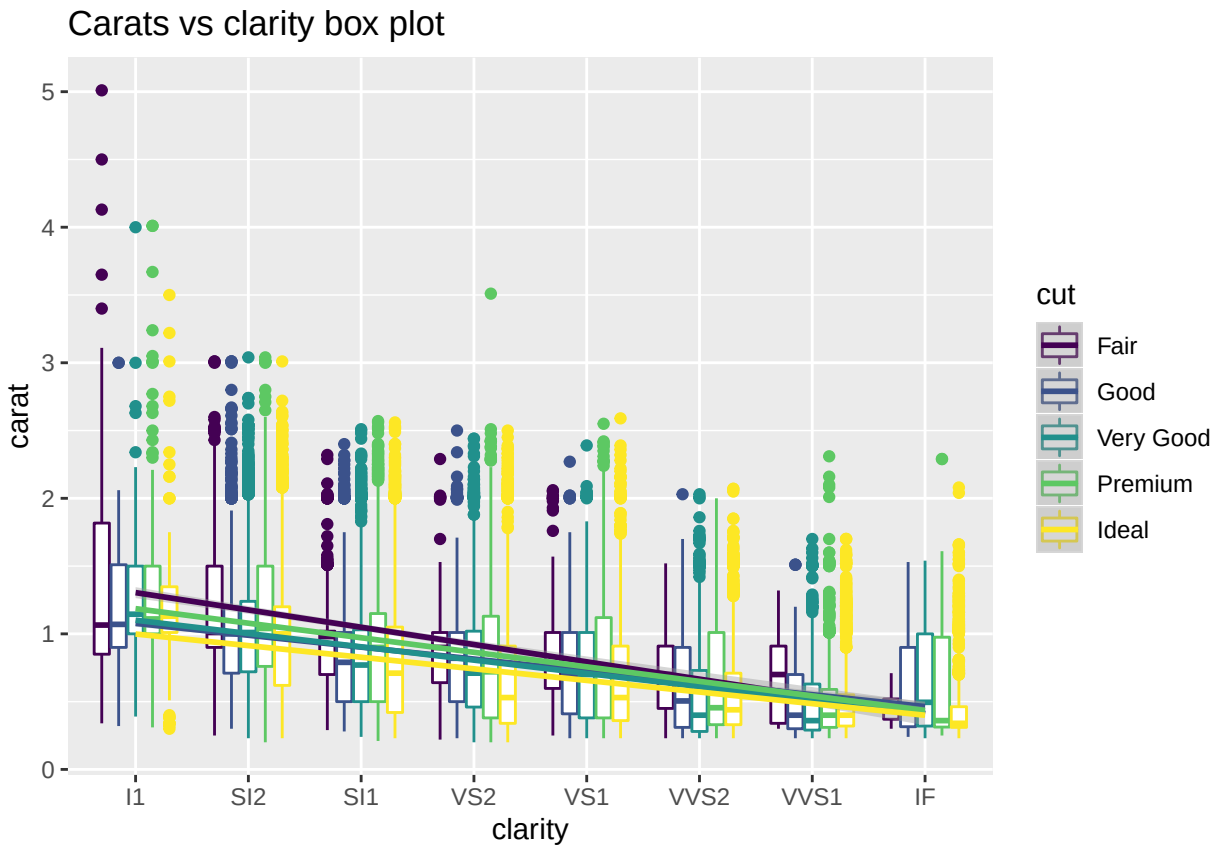

```
d1 + facet_wrap(~color, ncol=7)
```



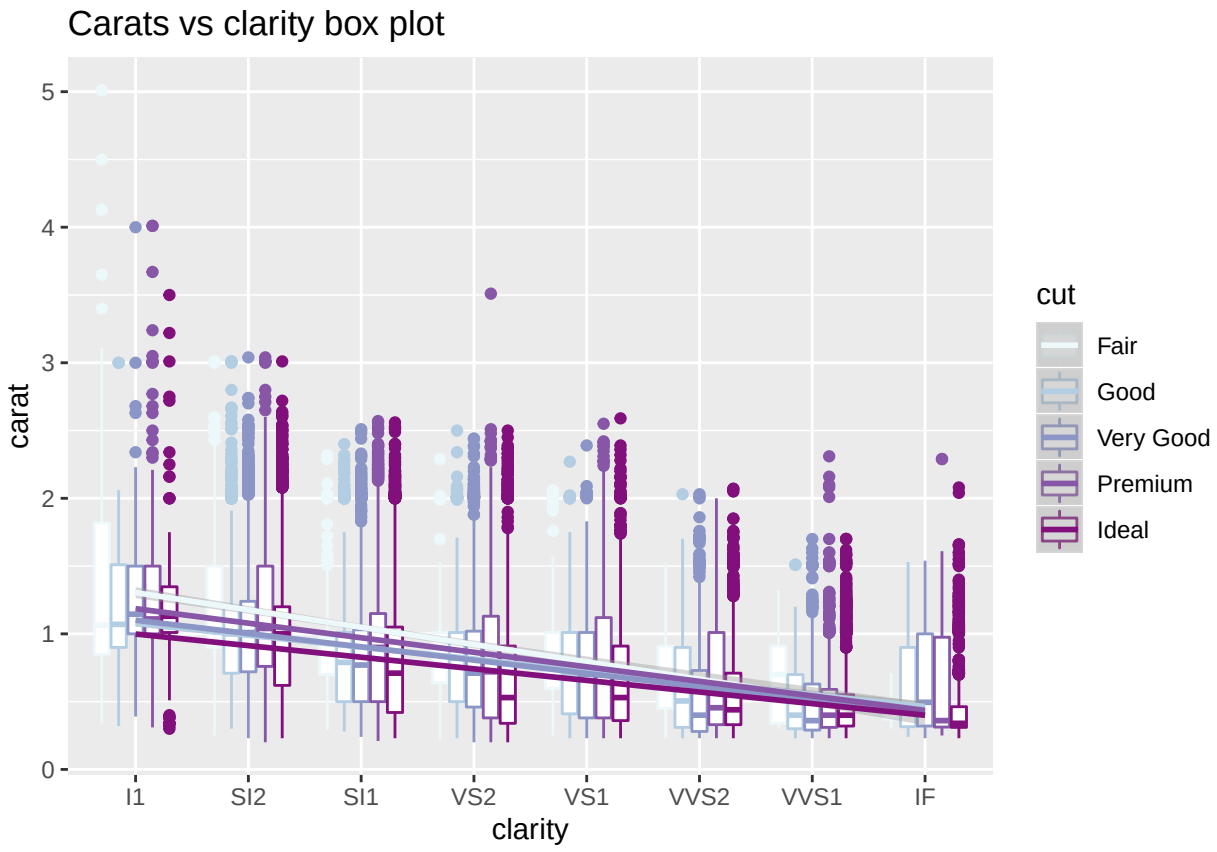

```
d1 + facet_grid(cut~color)
```




```
d1 + labs(title = "Carats vs clarity box plot", x="clarity",y="carat")
```




```
d2 <- d1 + labs(title = "Carats vs clarity box plot", x="clarity",y="carat") + scale_color_brewer(palet
d2
```

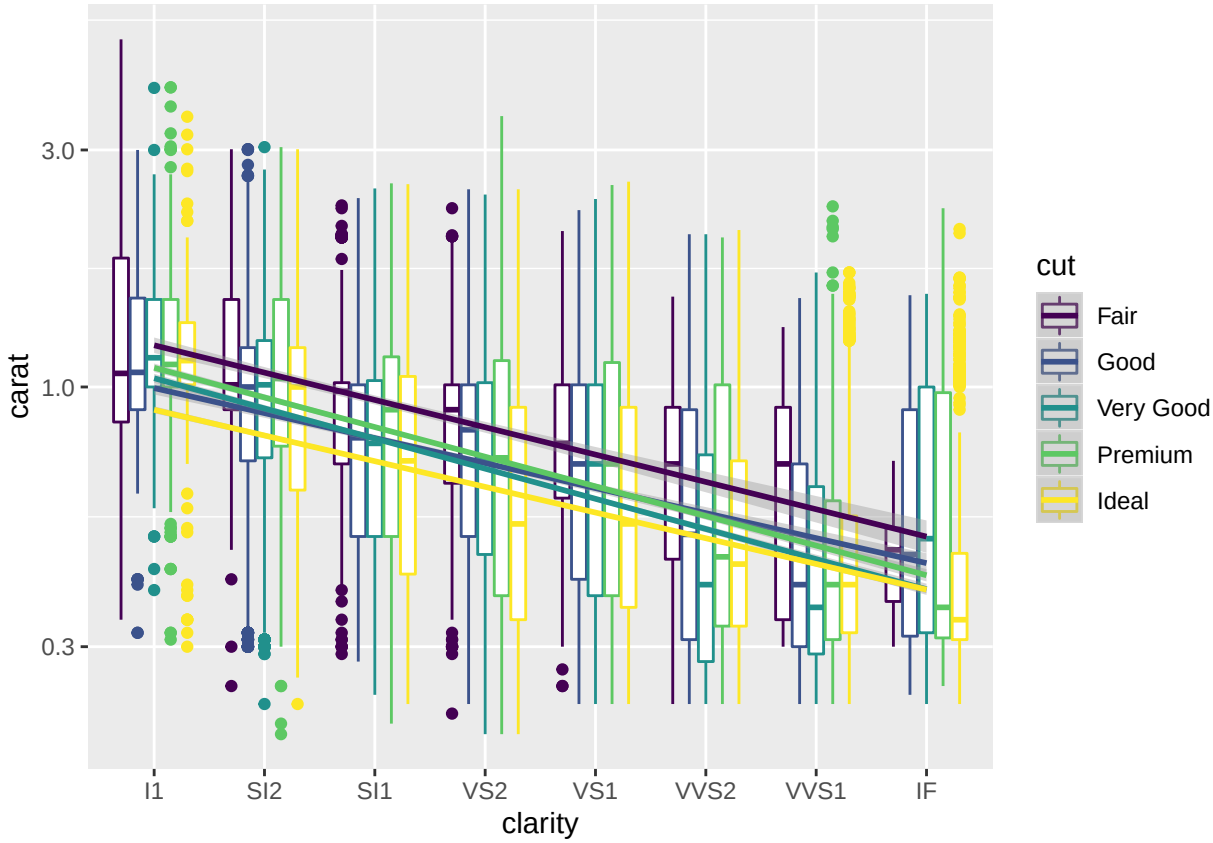



```
#save file
png(file="Carats vs clarity box plot.png", width = 500, height = 500)
d2
dev.off()
```

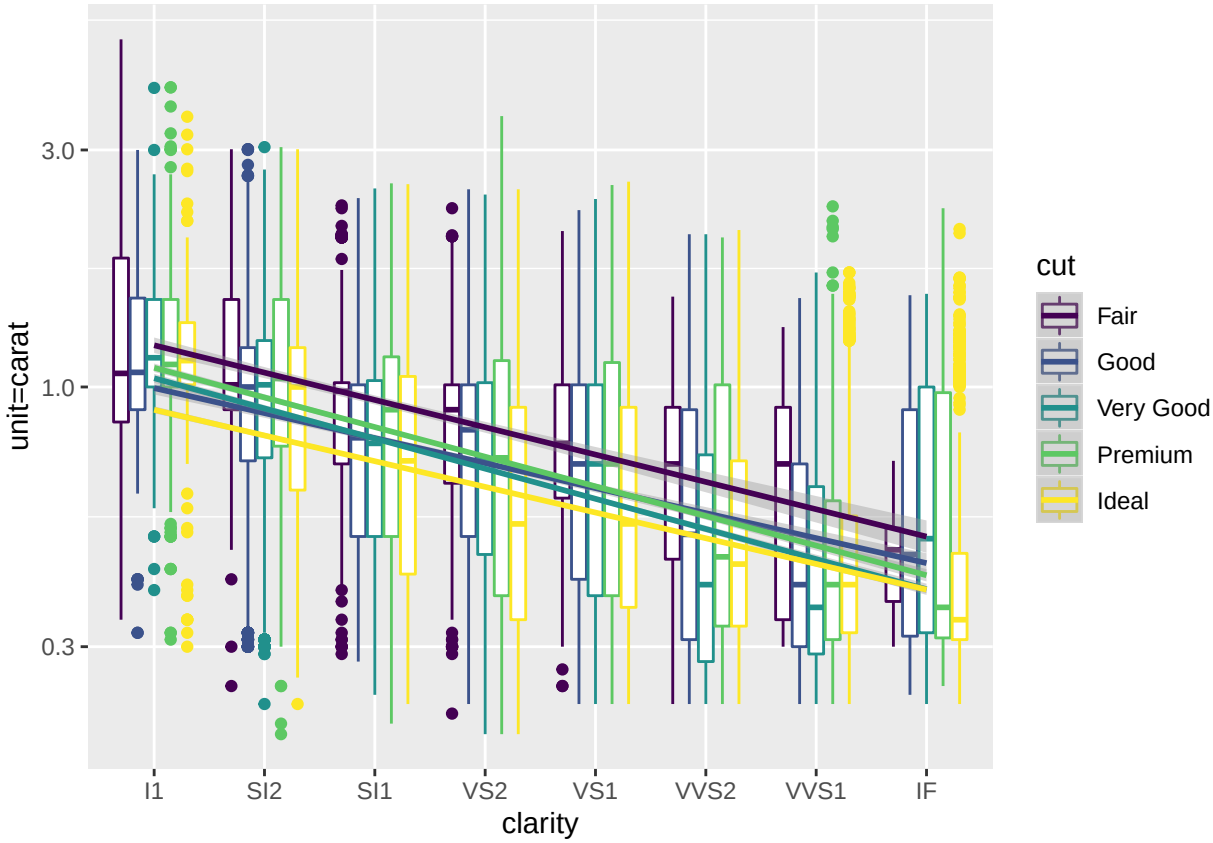
```
## pdf
##   2
```

4. Scale the y axis

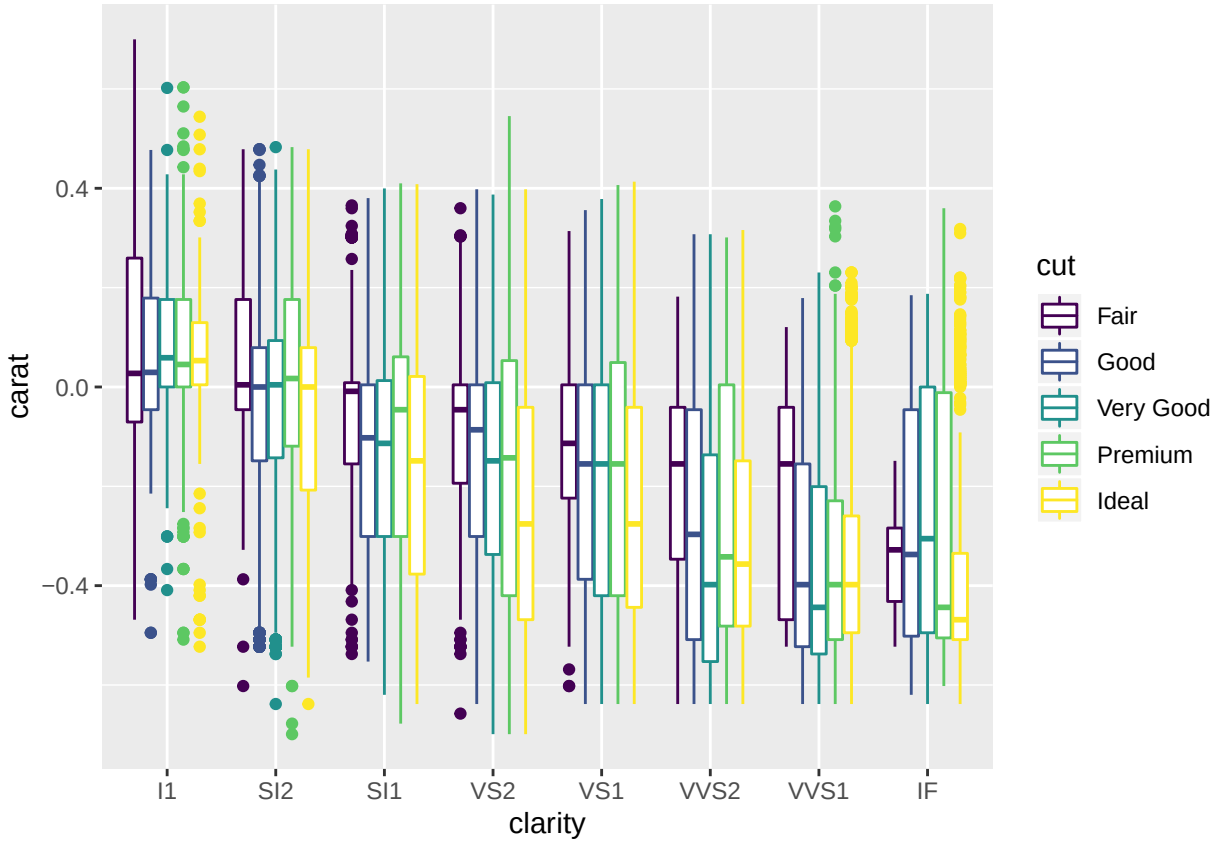
```
d1 + scale_y_continuous(trans = "log10" ) #what is the scale range? why?
```



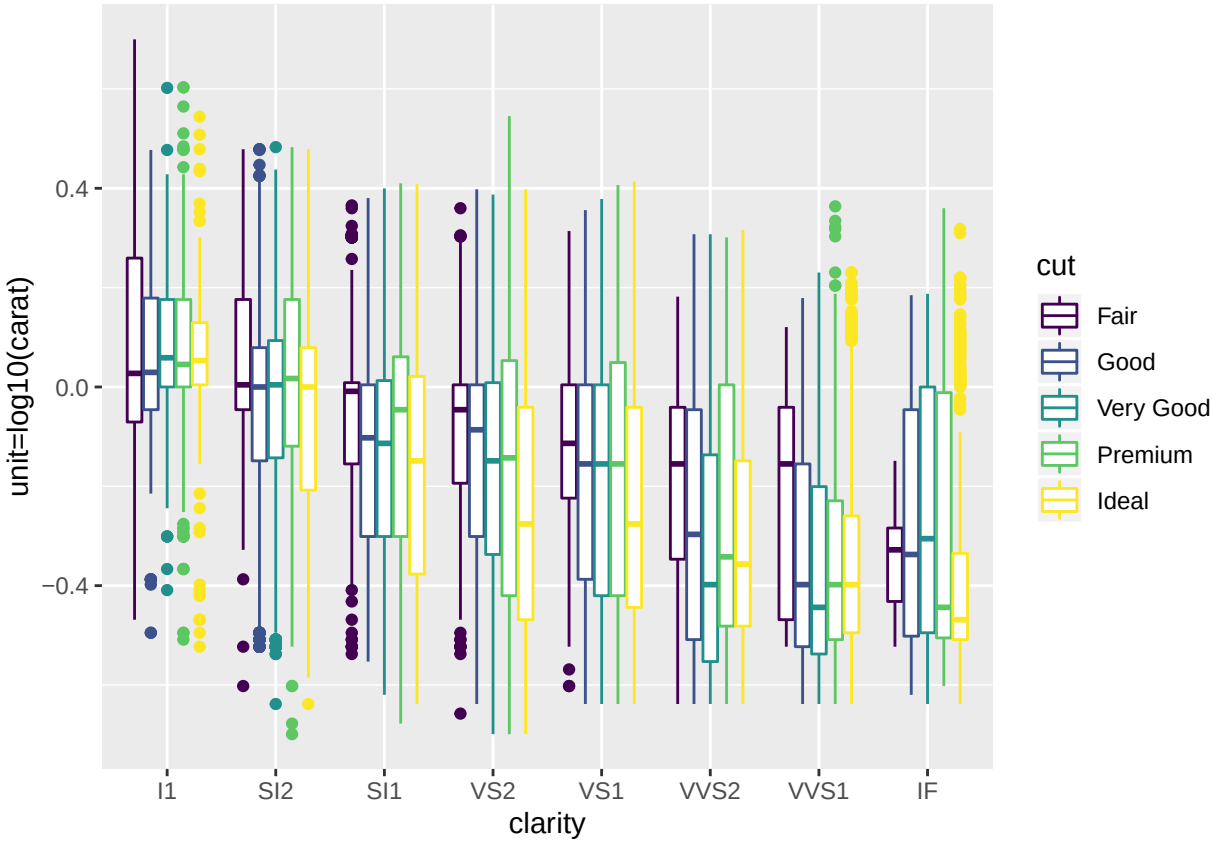

```
d1 + scale_y_continuous(trans = "log10" ) + ylab("unit=carat")
```



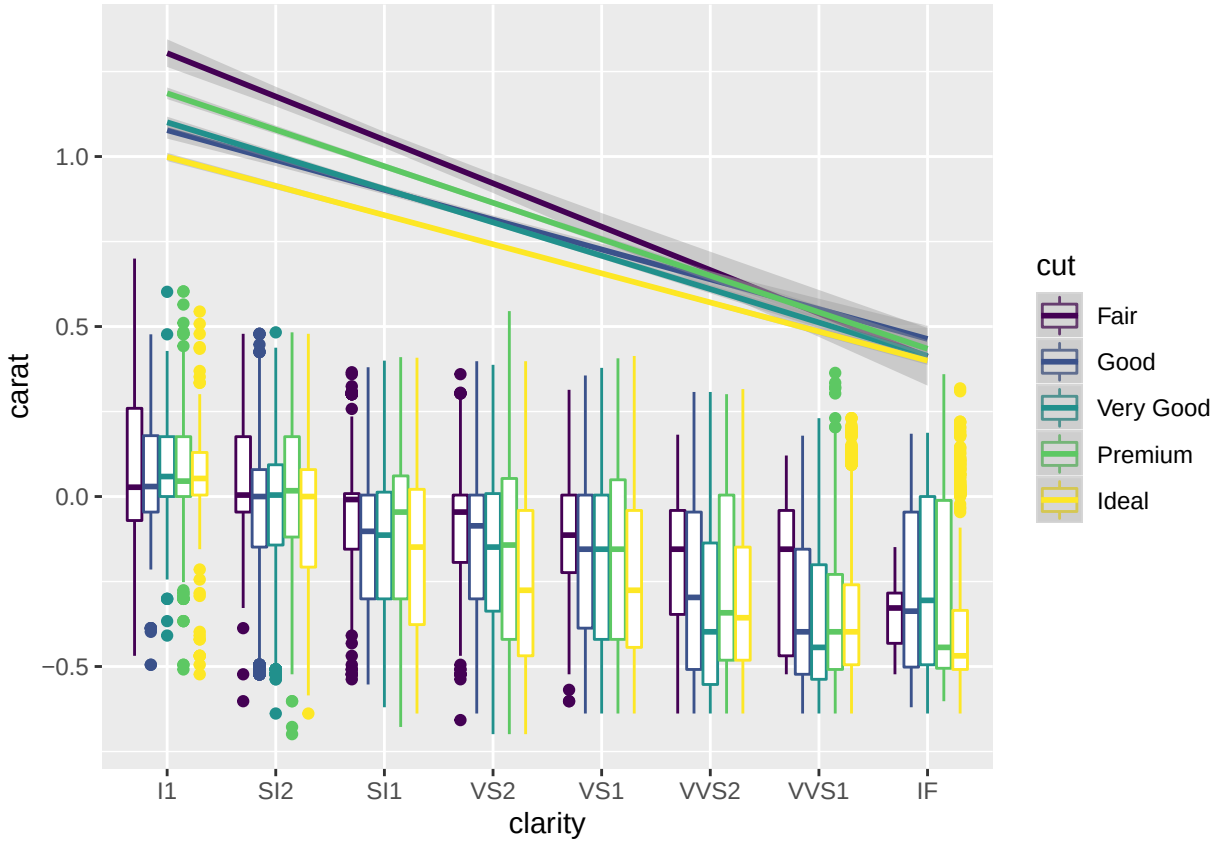

```
d + geom_boxplot(aes(clarity, log10(carat),colour=cut)) # what is the scale range now? what is the unit
```



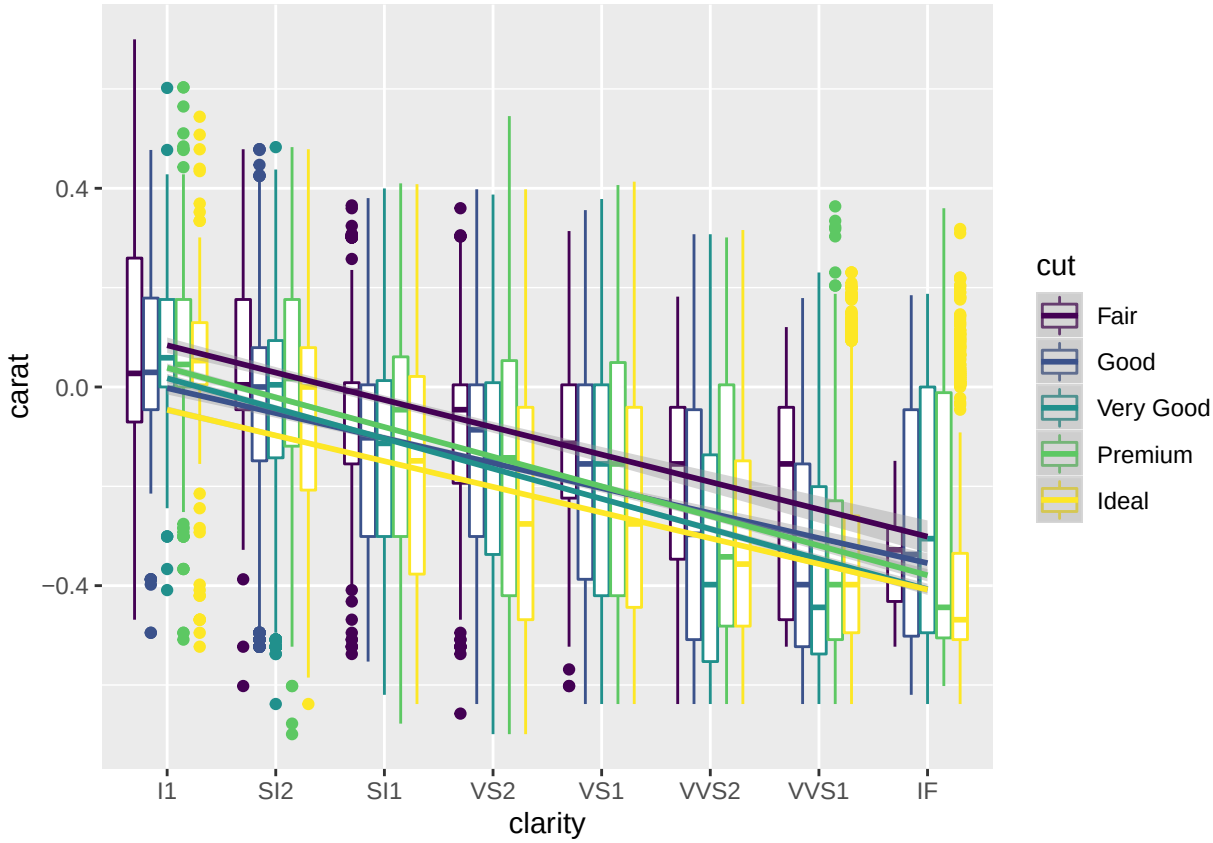

```
d + geom_boxplot(aes(clarity, log10(carat),colour=cut)) + ylab("unit=log10(carat)")
```



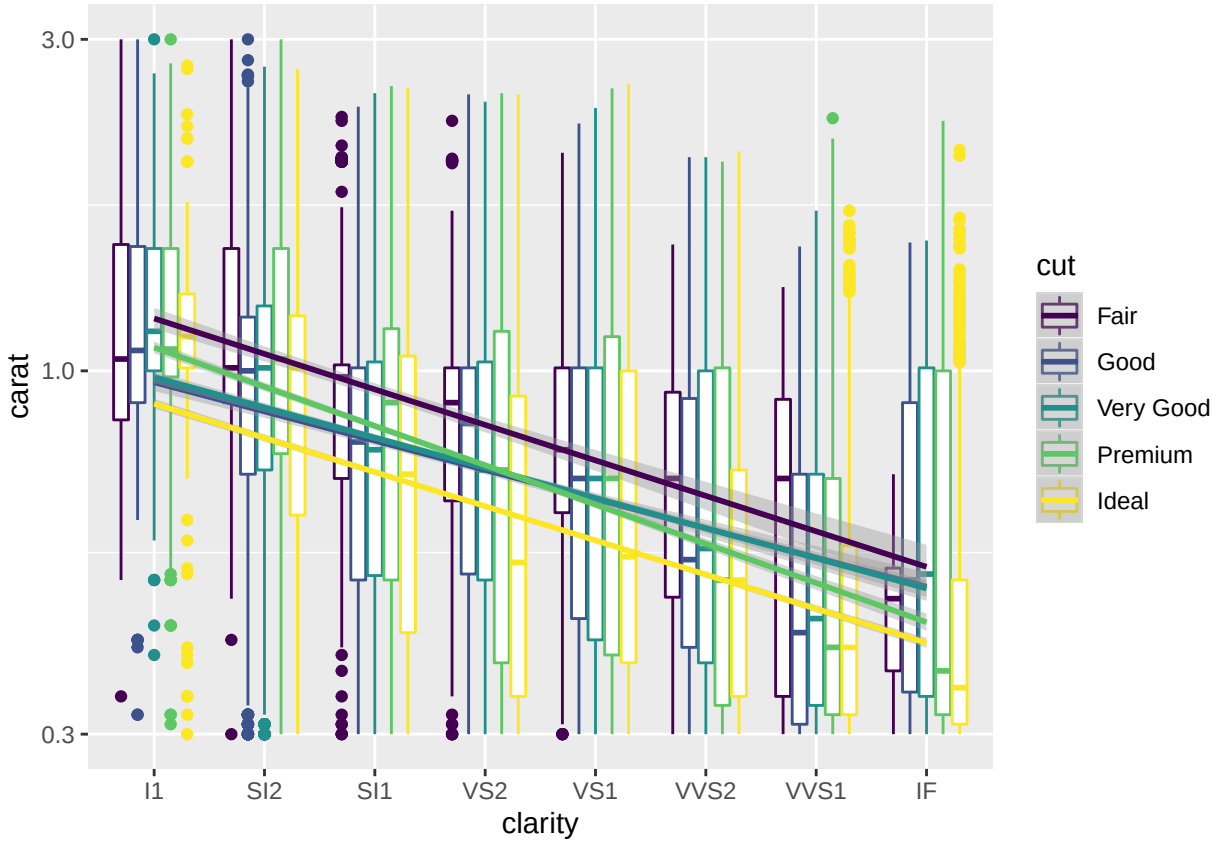

```
d + geom_boxplot(aes(clarity, log10(carat),colour=cut)) + sm
```




```
d + geom_boxplot(aes(clarity, log10(carat),colour=cut)) + geom_smooth(aes(y=log10(carat),group=cut,colou
```

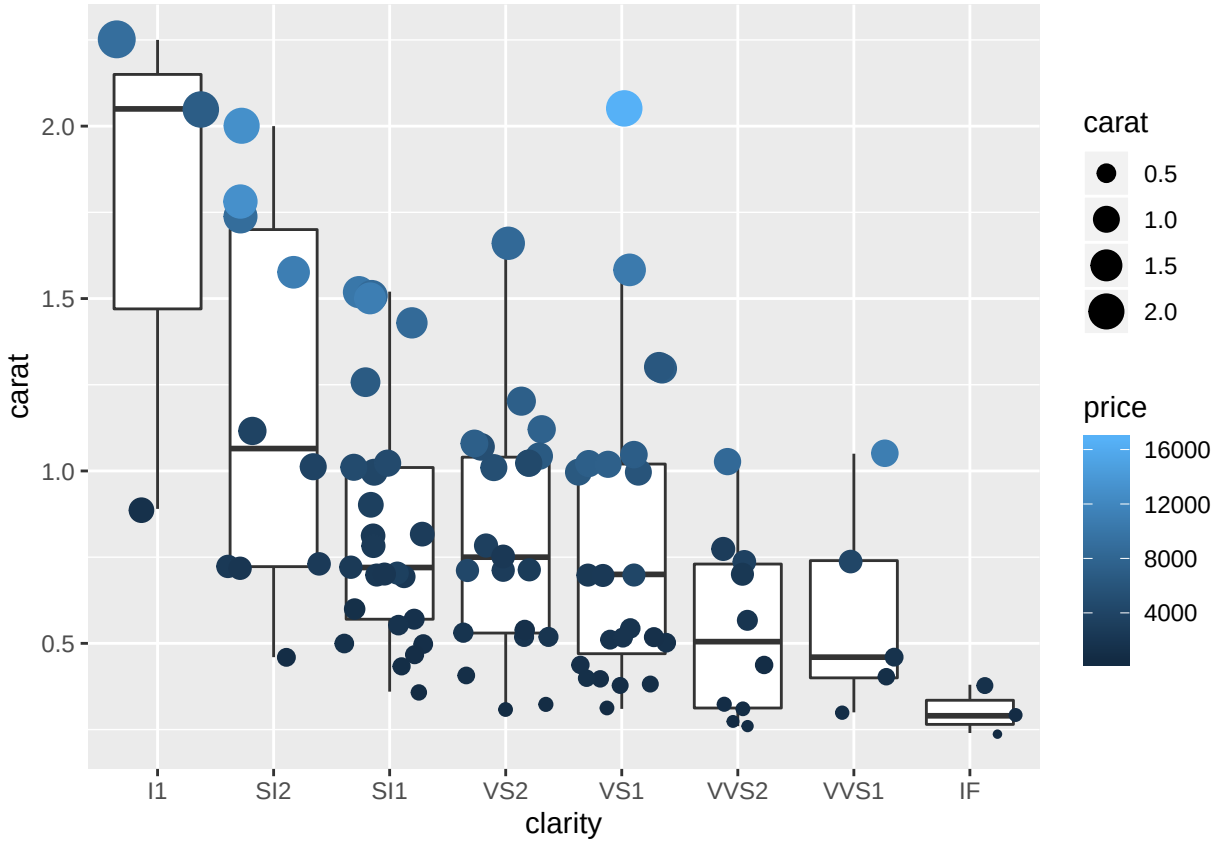



```
d1 + scale_y_continuous(trans = "log10" , limits = c(0.3, 3.0))
```

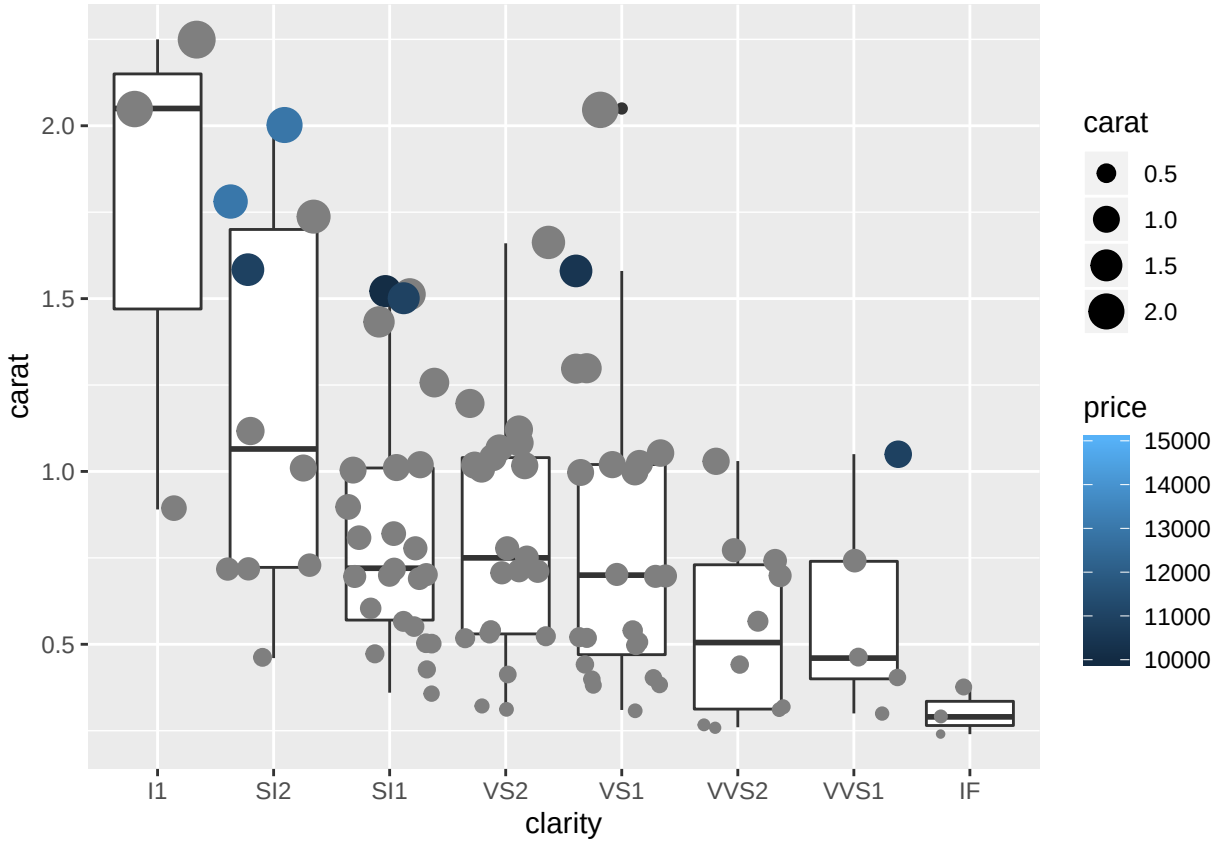


5. Jitter plot and scales

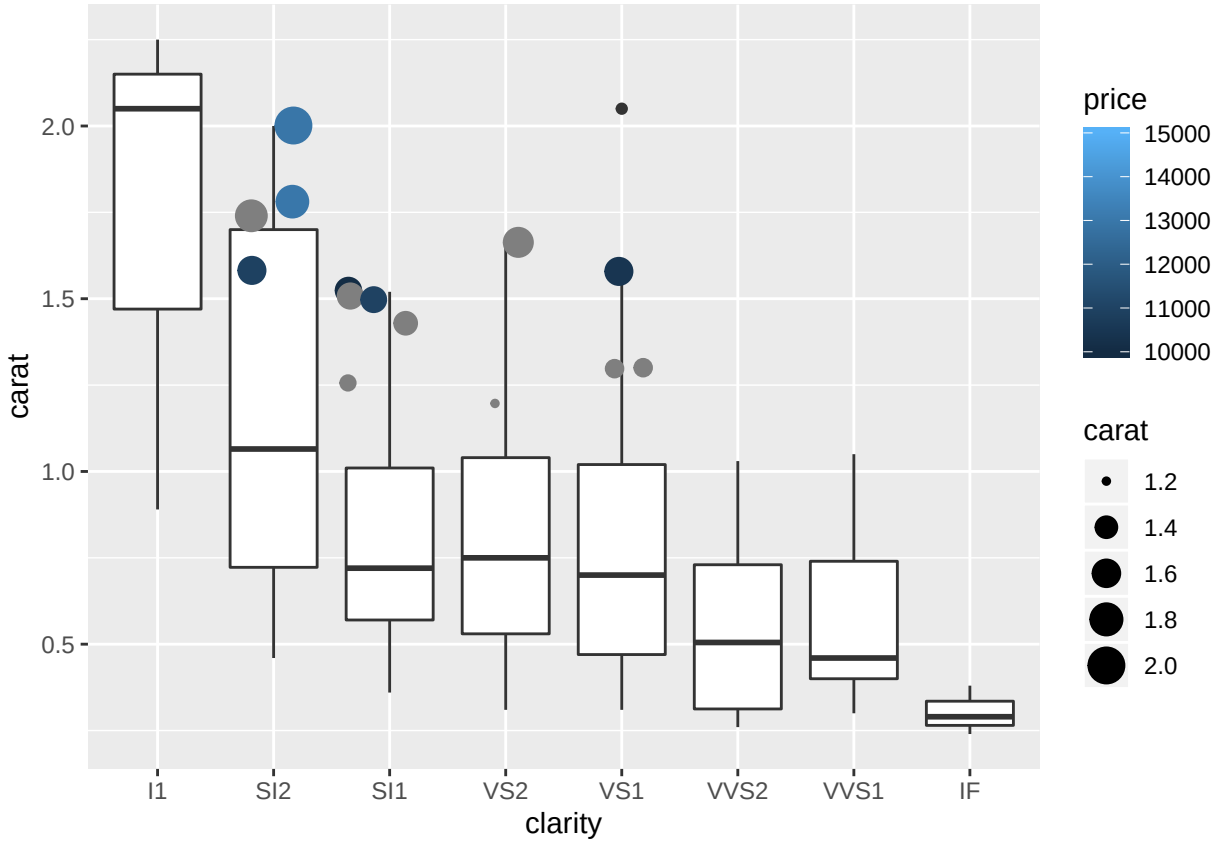
```
dia_sample <- diamonds[sample(nrow(diamonds),100,replace=F),]
ds <- ggplot(data=dia_sample, aes(x=clarity,y=carat))
ds + geom_boxplot() + geom_jitter(aes(size=carat,colour=price)) ##do you see anything related?
```




```
ds + geom_boxplot() + geom_jitter(aes(size=carat,colour=price)) + scale_color_gradient(limits= c(10000,
```

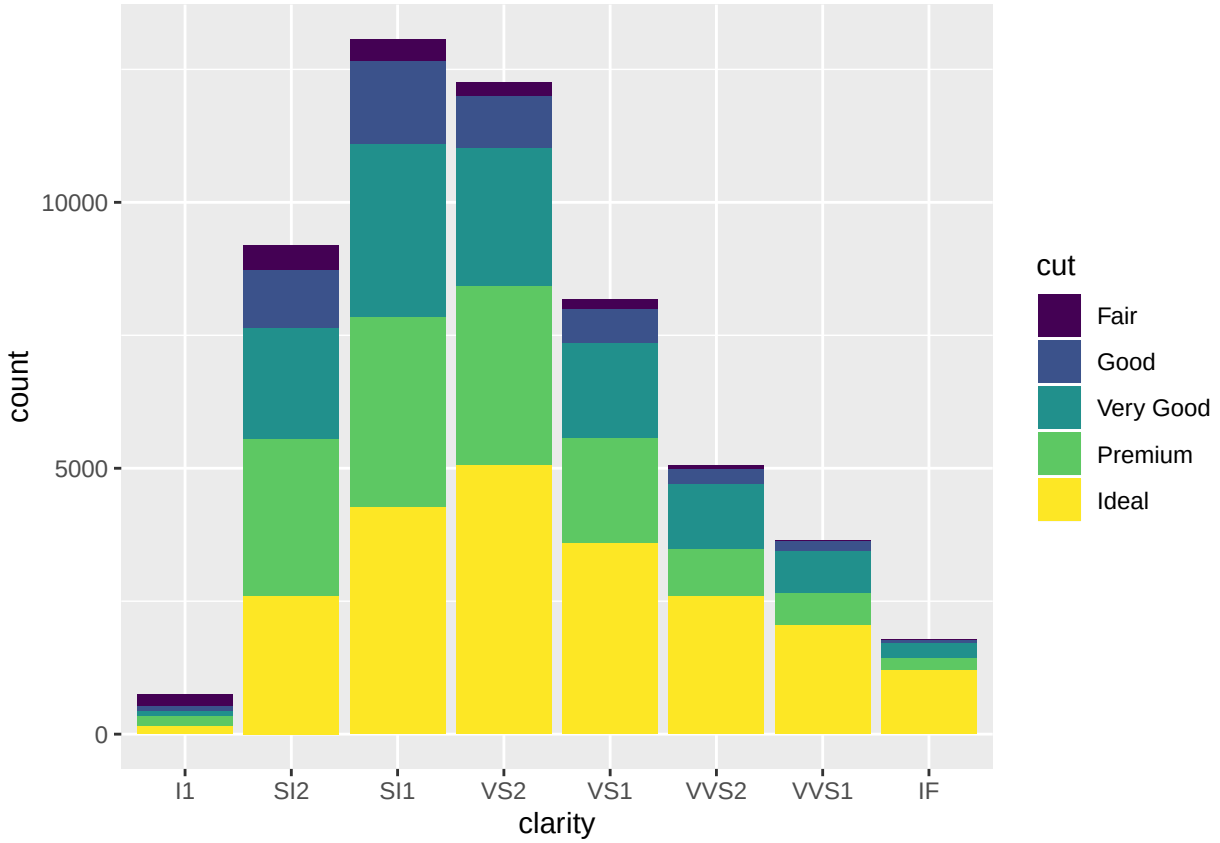



```
ds + geom_boxplot() + geom_jitter(aes(size=carat,colour=price)) + scale_color_gradient(limits= c(10000,
```

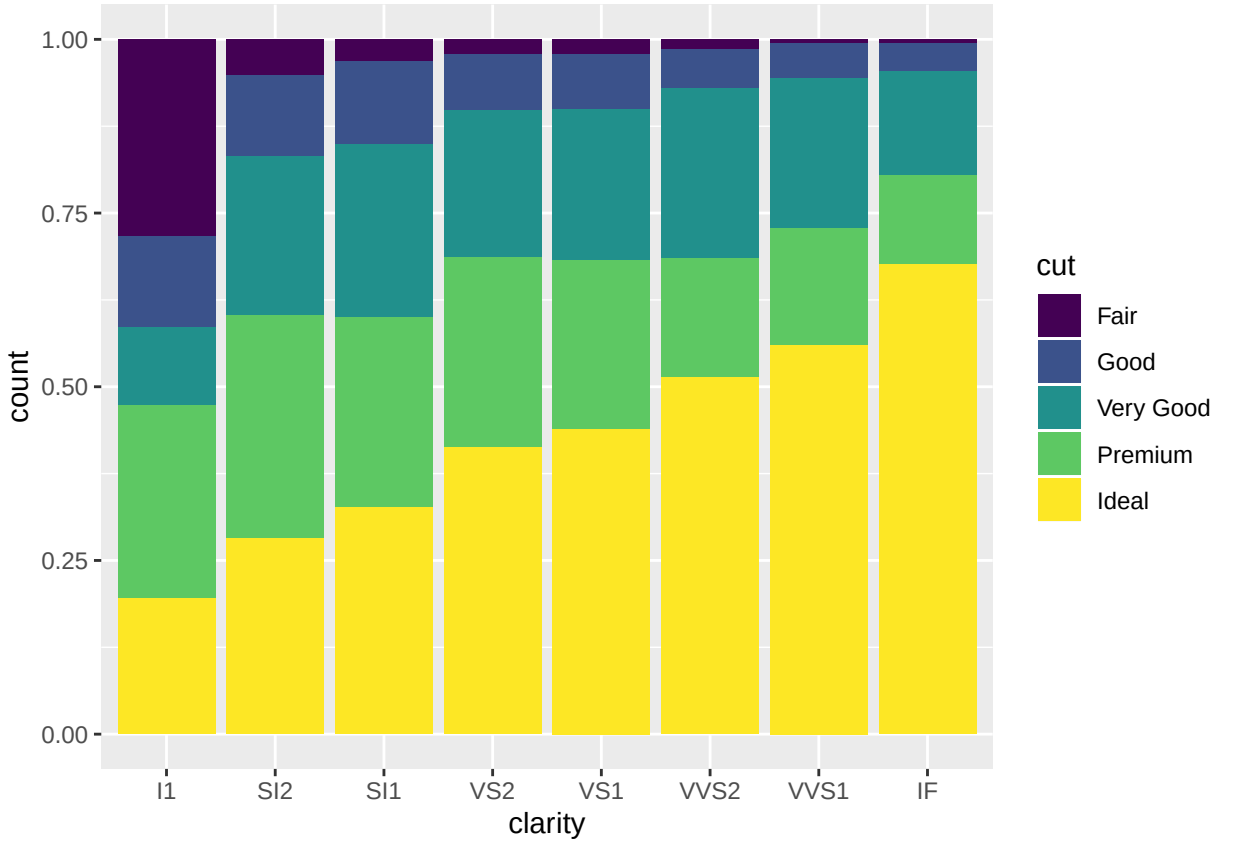


6. Position

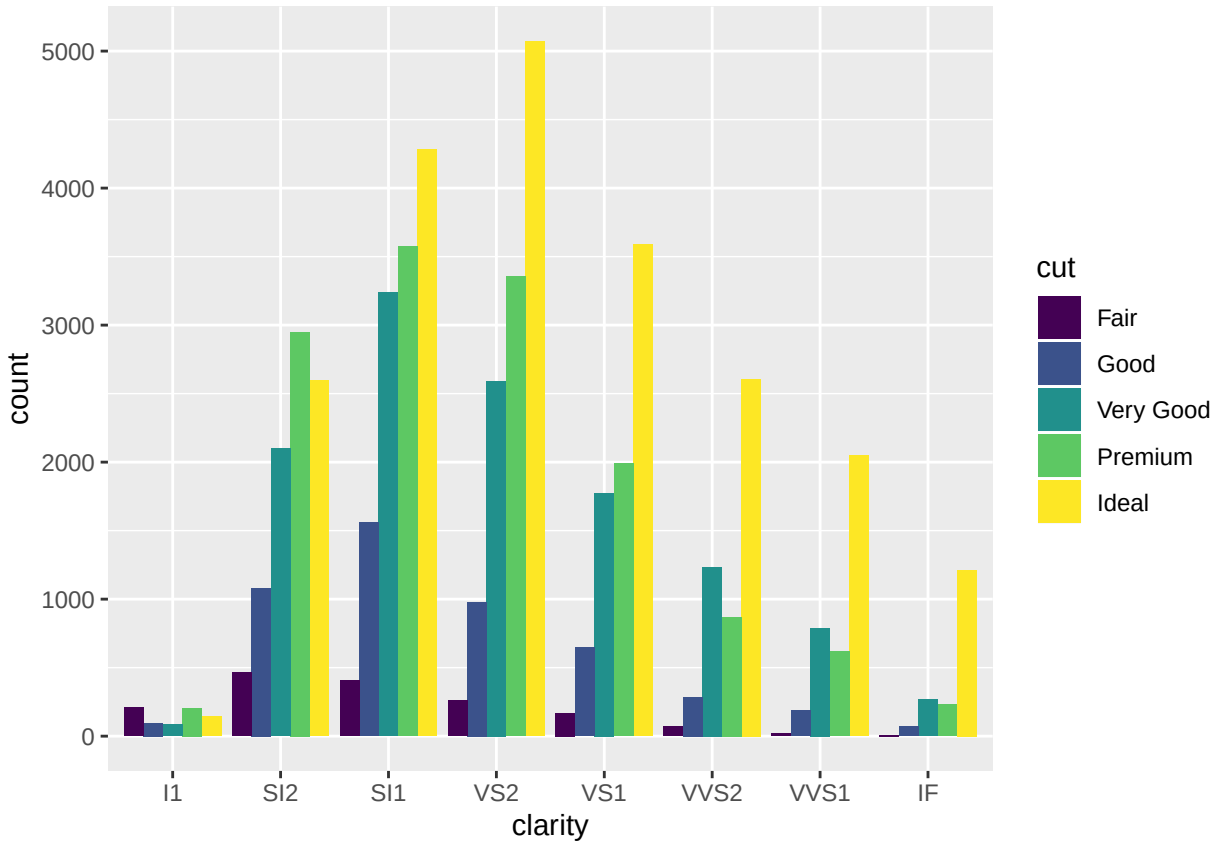
```
d <- ggplot(diamonds, aes(clarity))
d + geom_bar(aes(fill=cut),stat = "count", position = "stack")
```




```
d + geom_bar(aes(fill=cut),stat = "count", position = "fill")
```




```
d + geom_bar(aes(fill=cut),stat = "count", position = "dodge")
```


5000
4000
3000
2000
1000
0
count
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
clarity
cut
Fair
Good
Very Good
Premium
Ideal