

JPL-Caltech Virtual Summer School

Big Data Analytics

September 2 – 12, 2014

Ashish Mahabal
California Institute of Technology

R - VI

R graphics, comparison

- Base graphics (Ross Ihaka, S graphics driver)
 - Can not be modified
- Grid graphics (Paul Murrell, 2000)
 - Low level, but no support for statistical graphics
- Lattice package (Sarkar, 2008)
 - Based on grid. Detailed. Lacks a formal model
- ggplot2 (Hadley, 2005)
 - Layered, based on grammar of graphics, low level, static plots, extensions easy, flow like that of analysis
- ggobi, rggobi (Cook, Swayne, Wickham 2007,8)
 - Interactive, grammar of graphics
- **ggvis: New kid on the block**
 - browser based interactive graphics
 - works with shiny and RStudio

<http://cran.r-project.org/web/views/Graphics.html> (online list)

Grammar of graphics (Wilkinson, 2005)

- Mapping (***mappings***) from data (***data***) to aesthetic attributes of geometric objects (***geoms***).
- Including statistical transformations (***stats***) in specific coordinate systems
- Faceting (***facet***) to allow easy subsetting (synonyms: conditioning, latticing, trellising)
- ***scale, coord***

Hadley Wickham's ggplot2

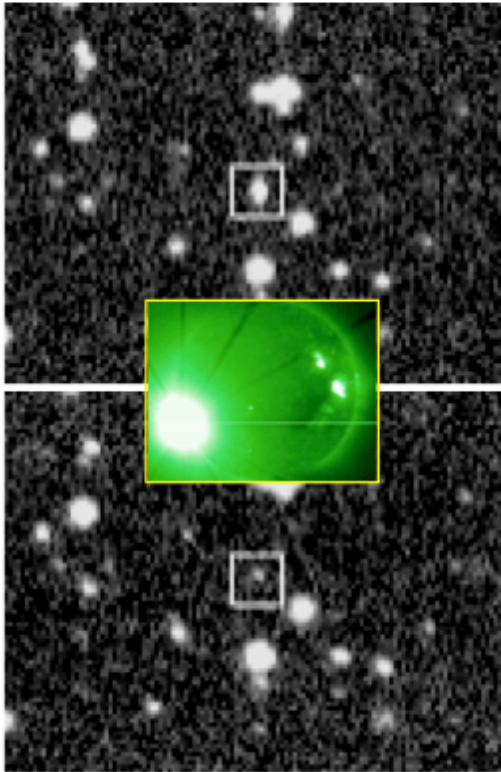
<http://ggplot2.org/>

- `install.packages("ggplot2")`
- `library(ggplot2)`

Lightcurve metrics for transients from CRTS

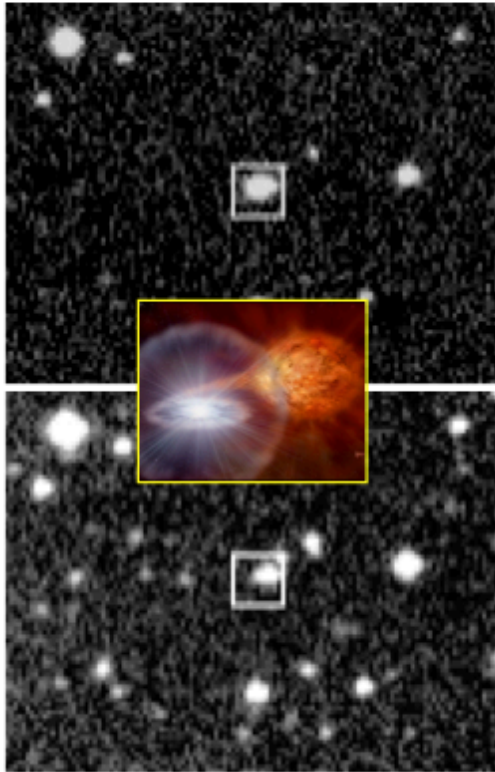
CSS090429:135125-075714

Flare star



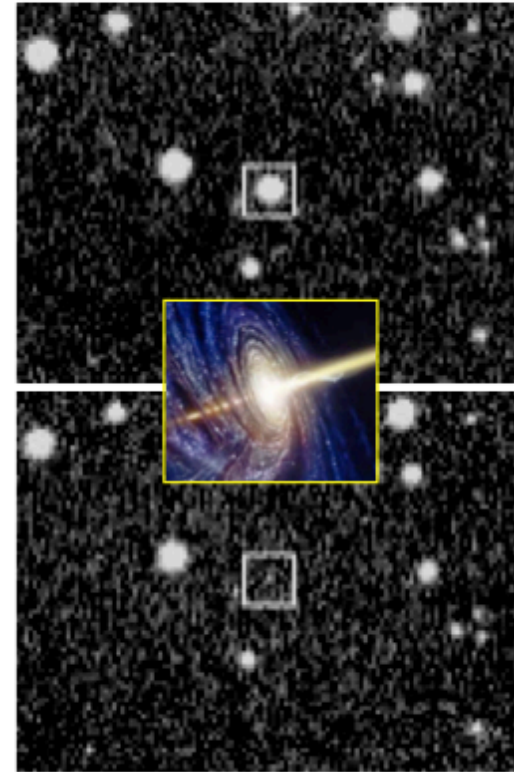
CSS090429:101546+033311

Dwarf Nova



CSS090426:074240+544425

Blazar, 2EG J0744+5438



Djorgovski

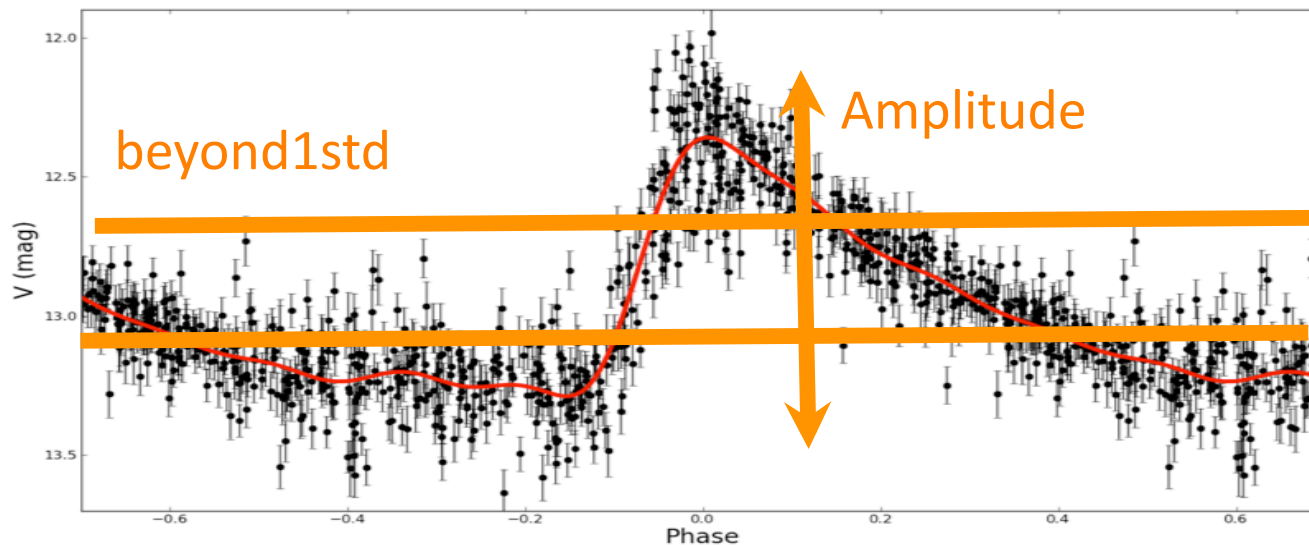
crts_6class

1619 rows; 20 useful variables; 1 categorical variable (6 classes)

Ashish Mahabal

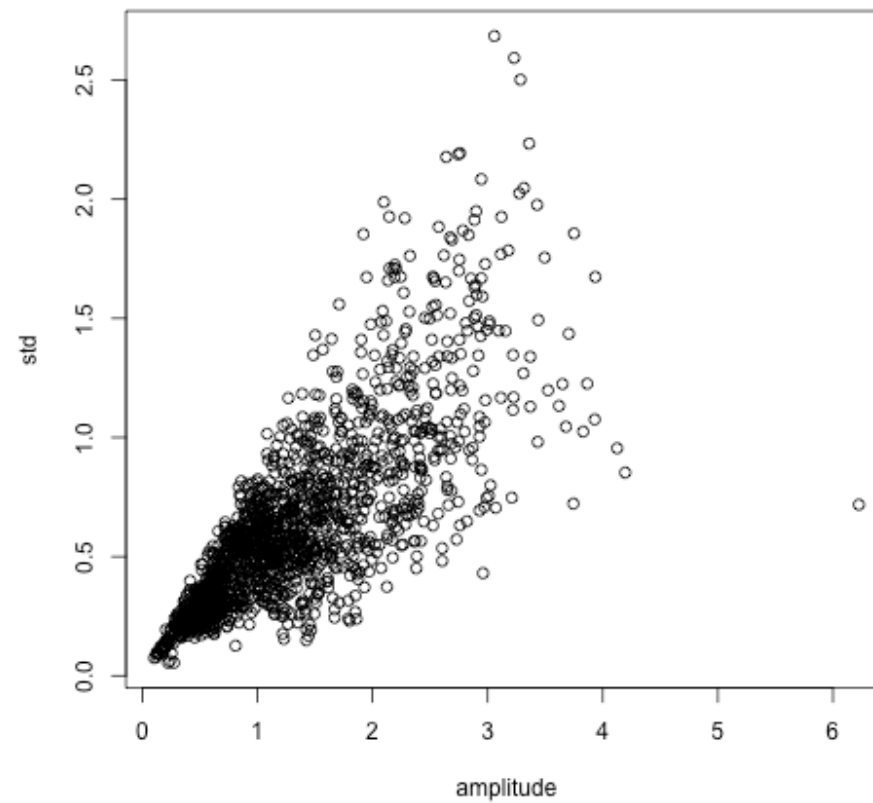
```
R> head(crts_6class, n=10)
```

	name	ra	dec	amplitude	...
1	CSS121123:051056-102401	77.73235	-10.400404	0.895000	
2	CSS121123:045020-093113	72.58246	-9.520167	0.750000	
3	CSS121120:020633+205707	31.63931	20.952106	2.375000	
4	CSS121114:093946+065210	144.94174	6.869316	1.495698	
5	CSS121114:011948-241624	19.94971	-24.273572	0.281392	
6	CSS121112:211504-183405	318.76524	-18.568049	0.603089	
7	CSS121025:180049+523235	270.20424	52.543164	2.065000	
8	CSS121014:224648+065635	341.69902	6.942954	1.880000	
9	CSS121011:225648-274325	344.19937	-27.723736	1.040000	
10	CSS121010:174747+552918	266.94669	55.488411	2.280000	



Scatter plots

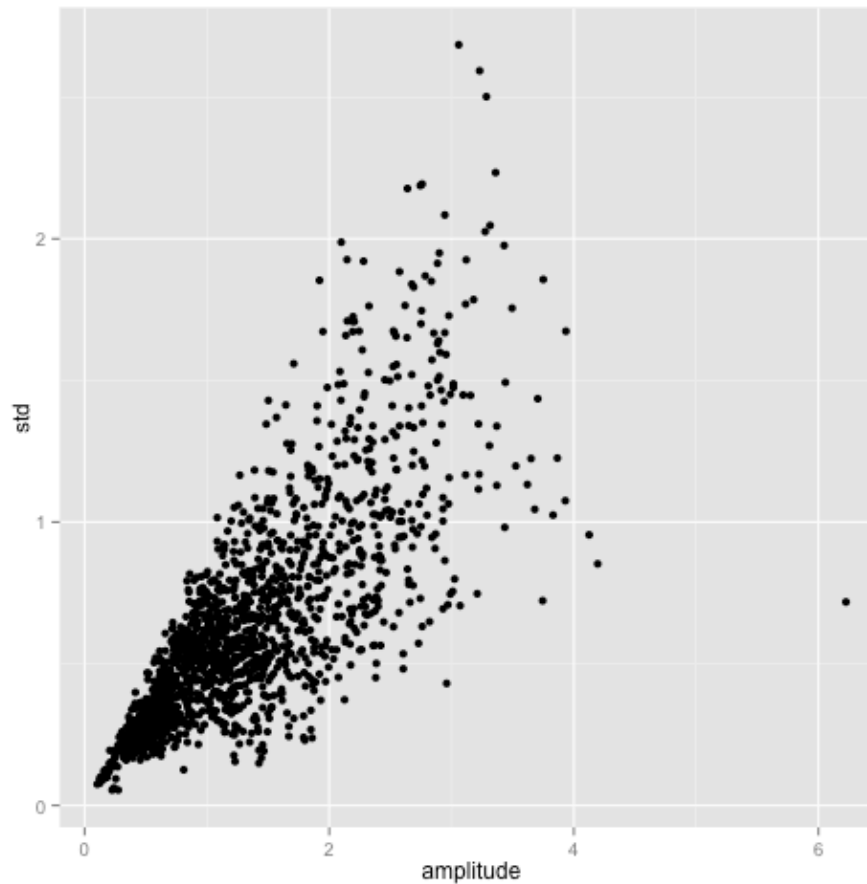
```
plot(amplitude,std)
```



Scatter plots

```
plot(amplitude,std)
```

```
qplot(amplitude,std) # Though data=crts_6class may have to be specified once
```



Scatter plots

```
plot(amplitude,std)
```

```
qplot(amplitude,std, data=crts_6class)
```

```
ggplot(aes(amplitude,std)) # data=crts_6class has to be specified
```

```
ggplot(data=crts_6class, aes(amplitude,std)) # still nothing, because  
there is no rendering instruction yet!
```

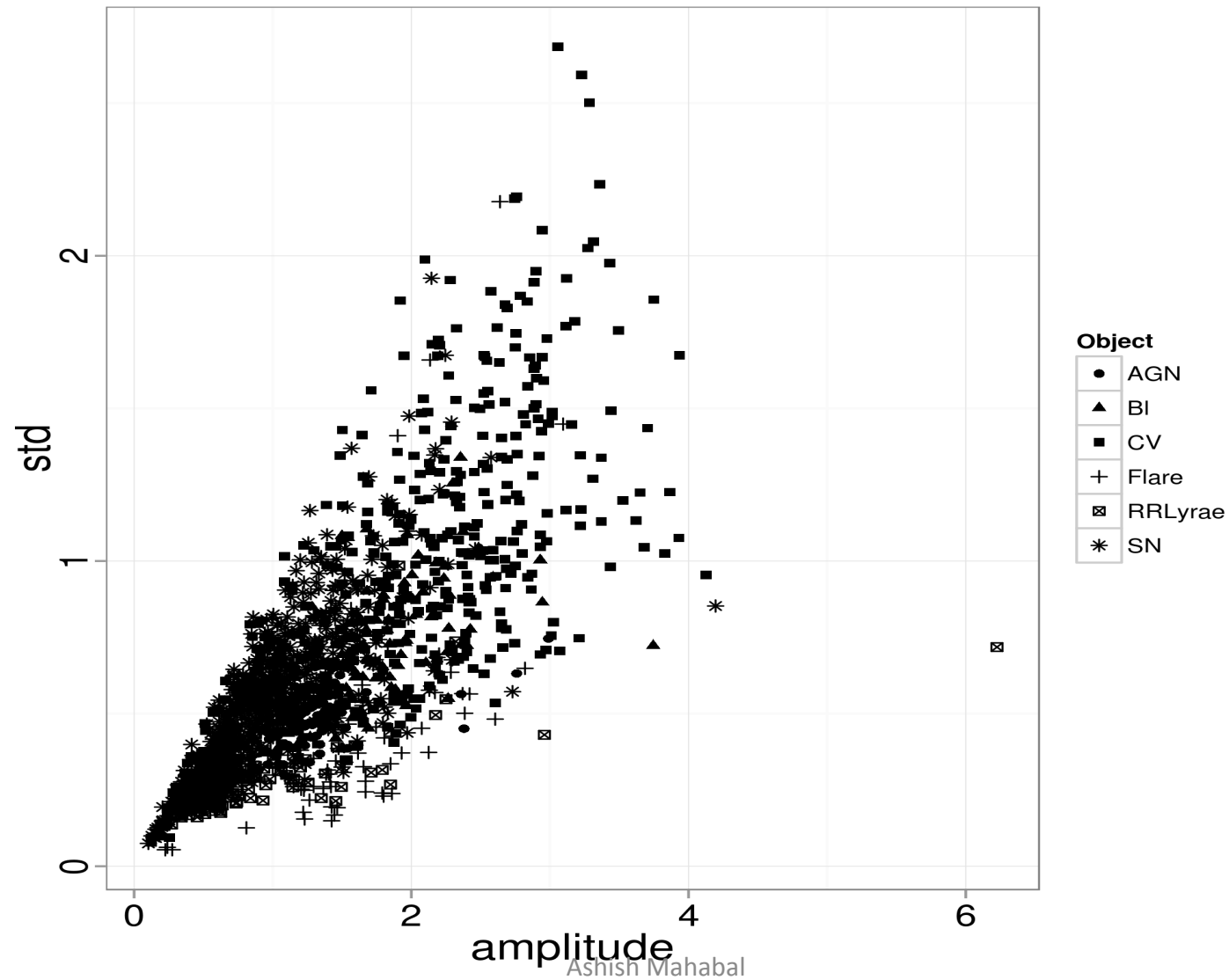
```
ggplot(data=crts_6class, aes(amplitude,std)) + geom_point() # Finally!
```

Scatter plots

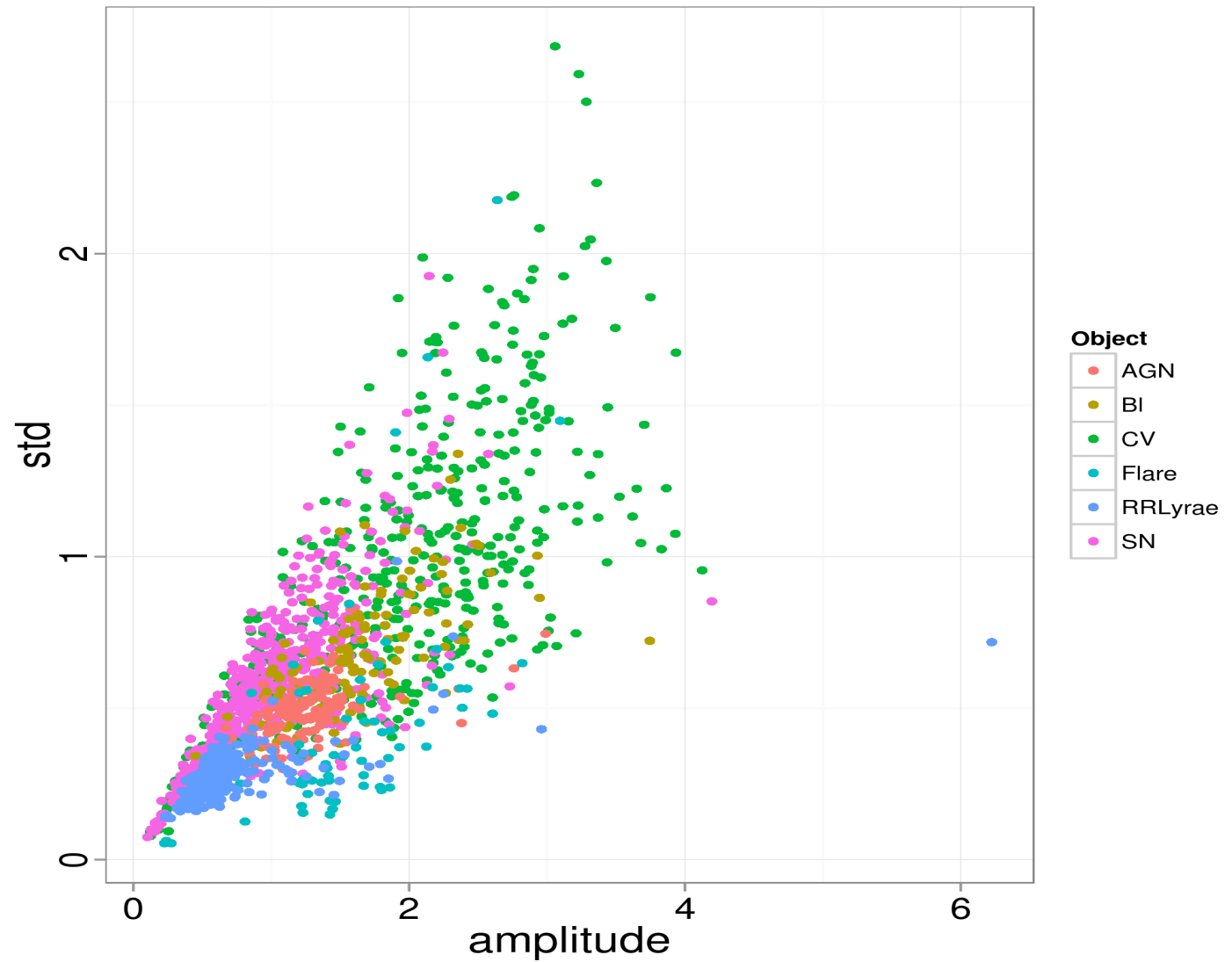
Alternately (the layering aspect):

```
R> p <- ggplot(data=crts_6class, aes(amplitude,std))      # Nothing  
R> p + geom_point()                                   # Voila!
```

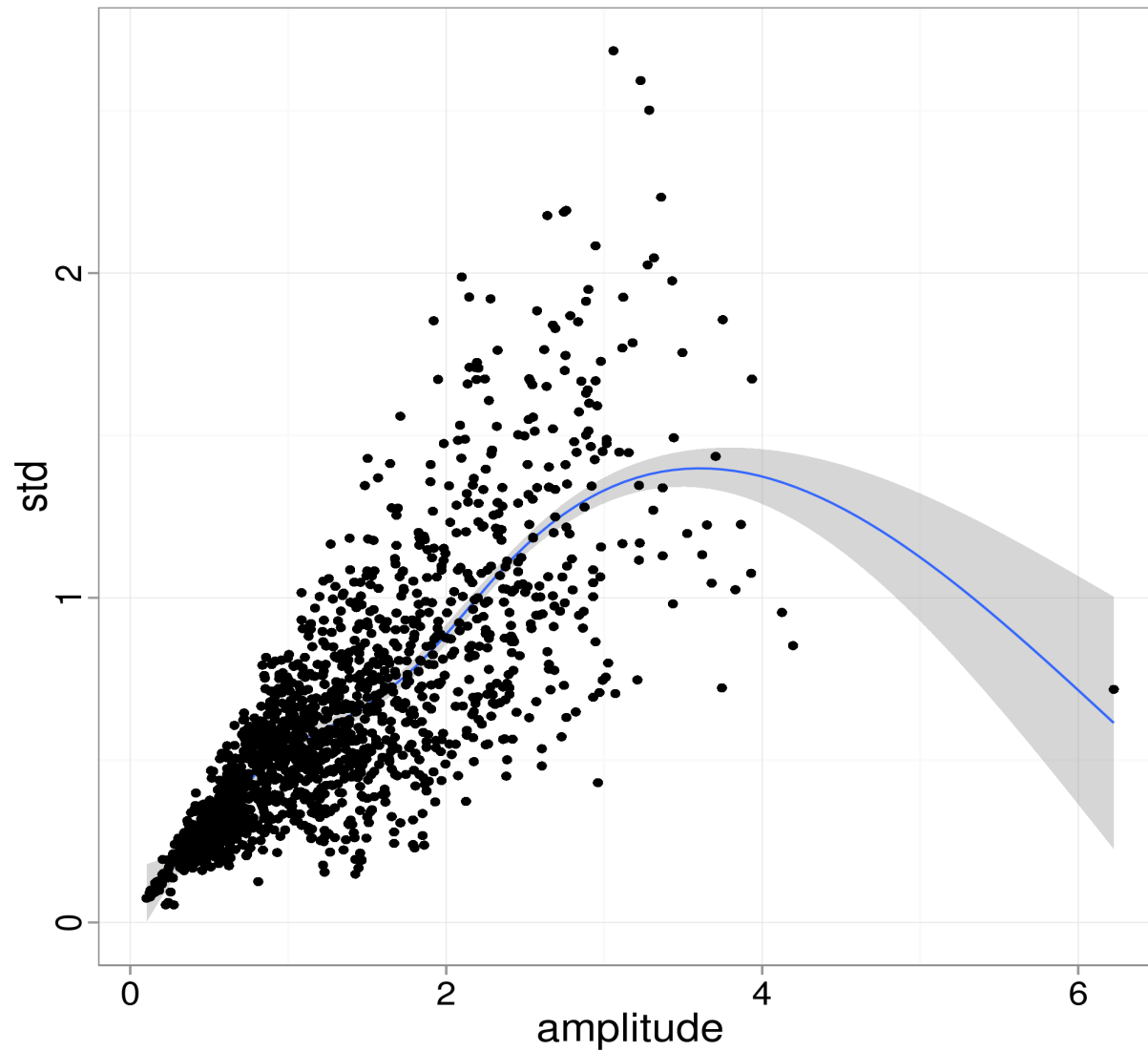
```
ggplot(data=crts_6class, aes(x=amplitude, y=std, shape=object)) +  
geom_point()
```



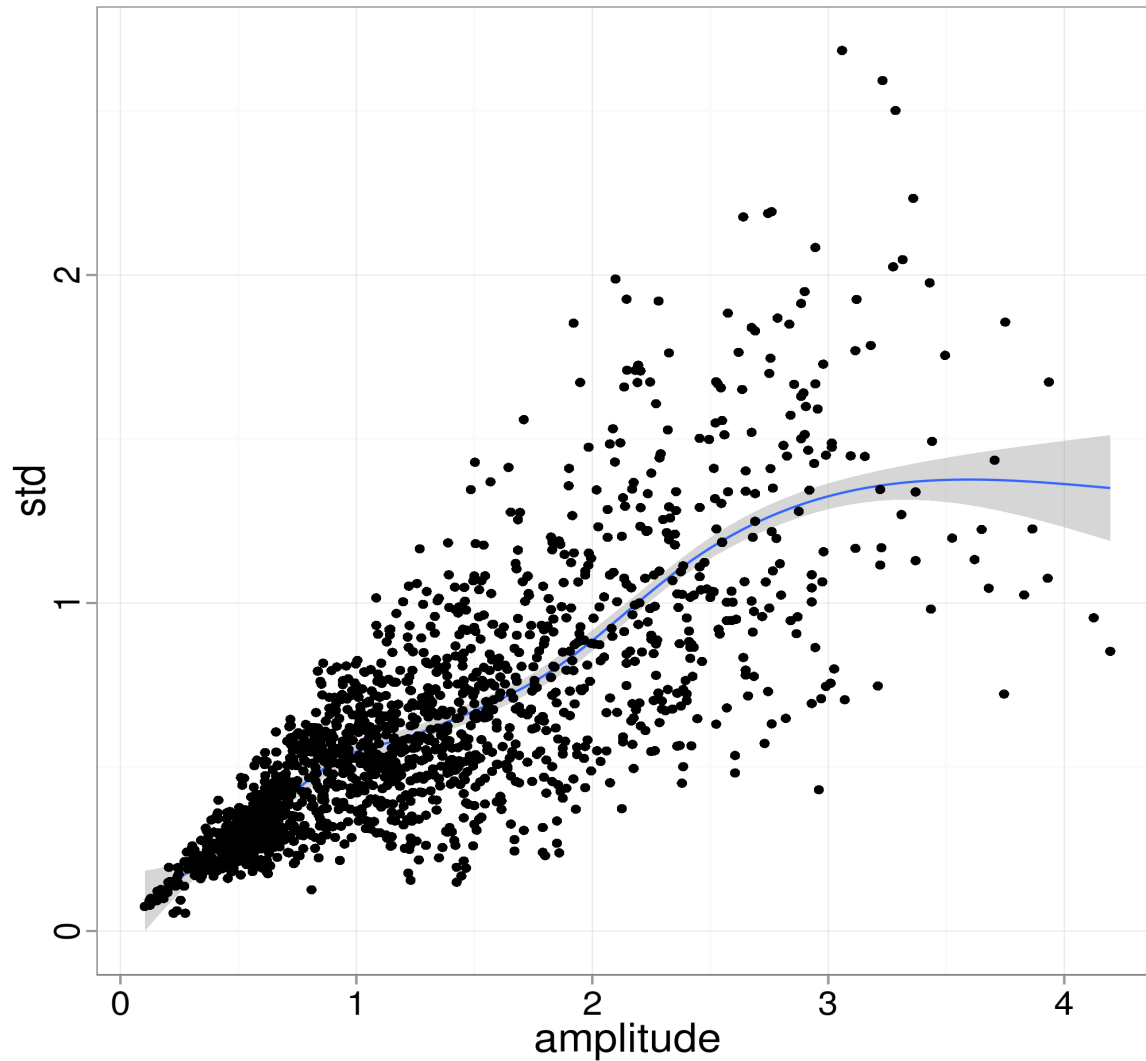
```
ggplot(data=crts_6class, aes(x=amplitude, y=std, colour=object)) +  
geom_point()
```



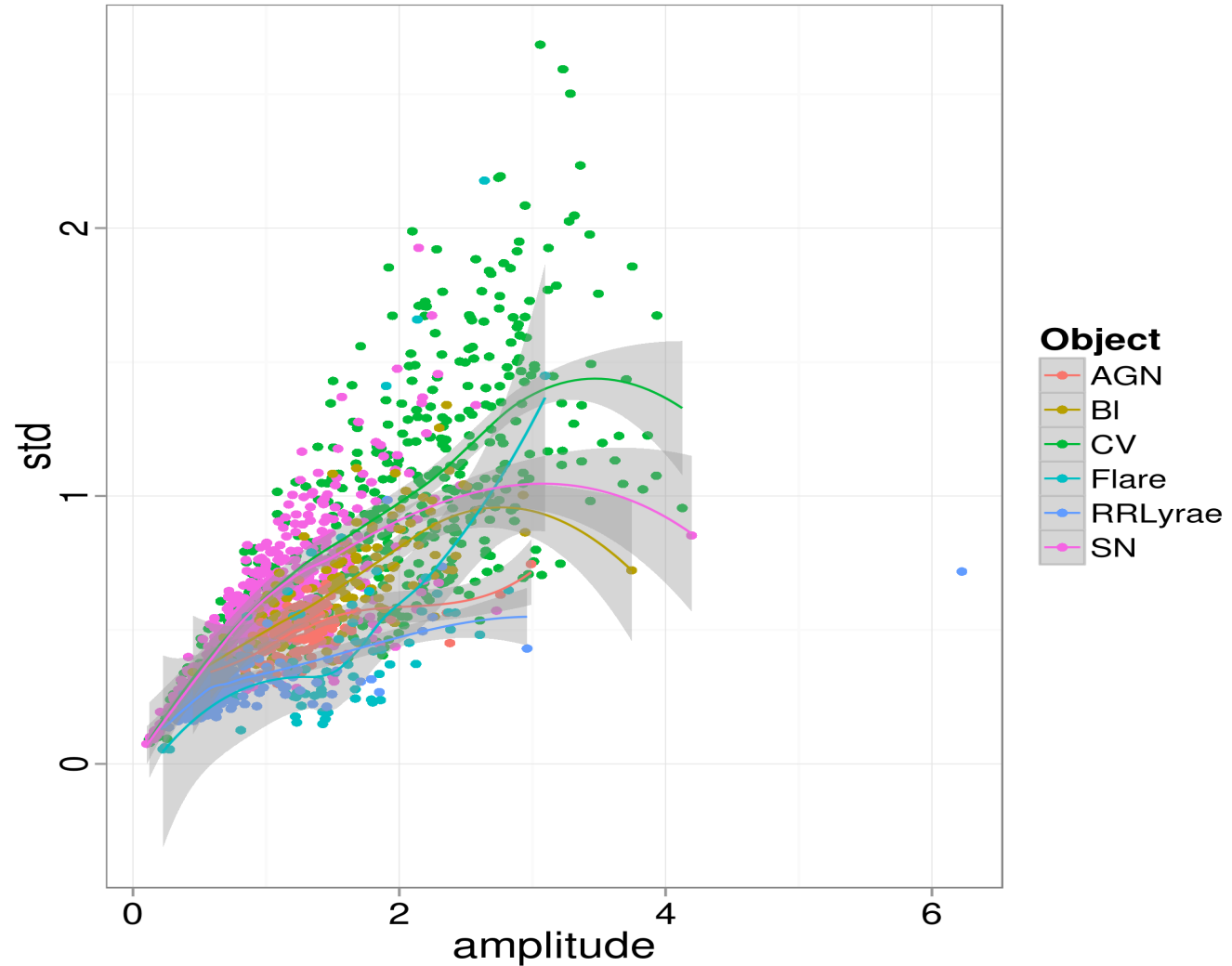
```
ggplot(data=crts_6class, aes(x=amplitude, y=std)) + geom_point() +  
geom_smooth()
```



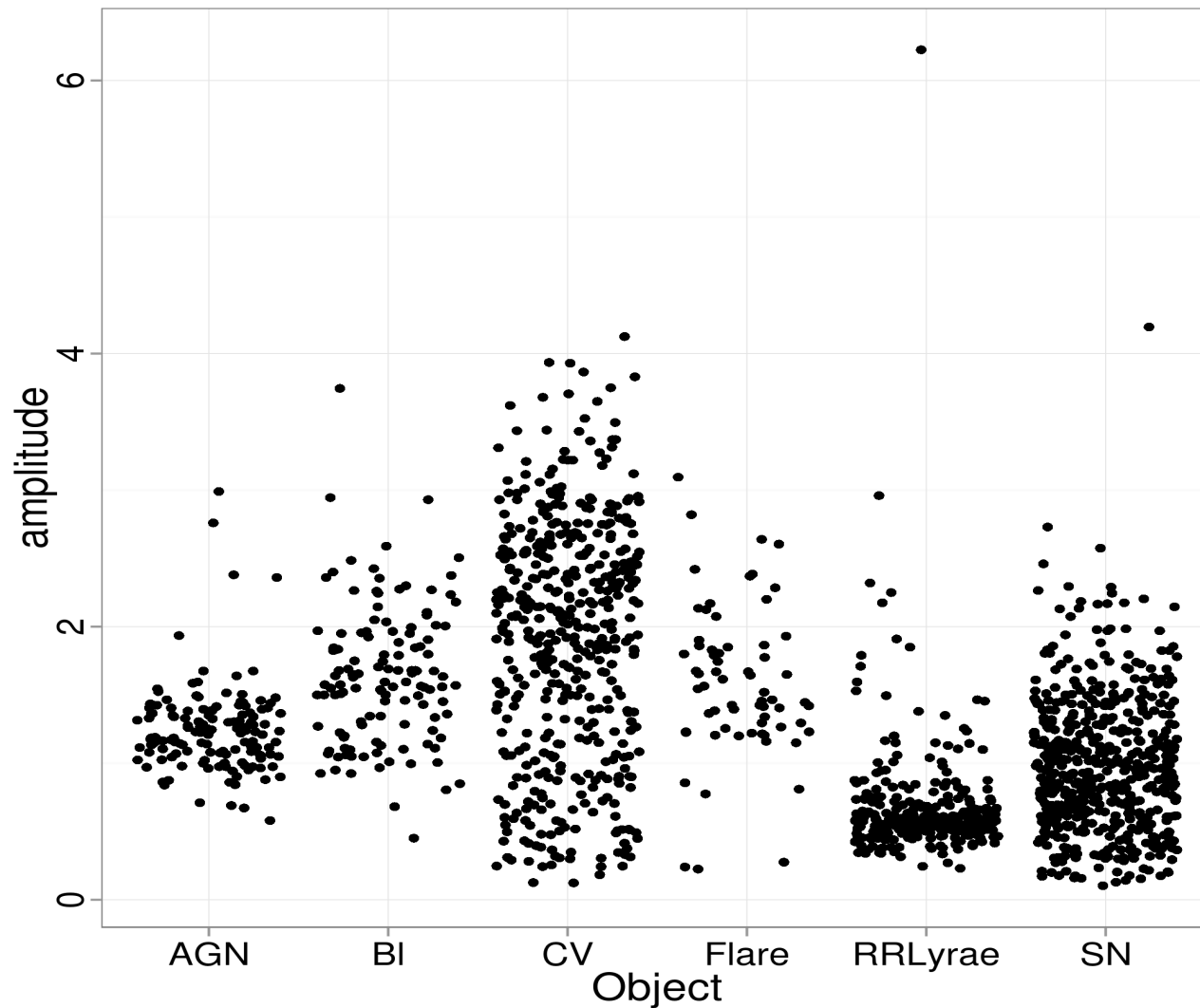
```
ggplot(data=crts_6class, aes(x=amplitude, y=std)) + geom_point() +  
geom_smooth()
```



```
ggplot(data=crts_6class, aes(x=amplitude, y=std, colour=object))  
+ geom_point() + geom_smooth()
```

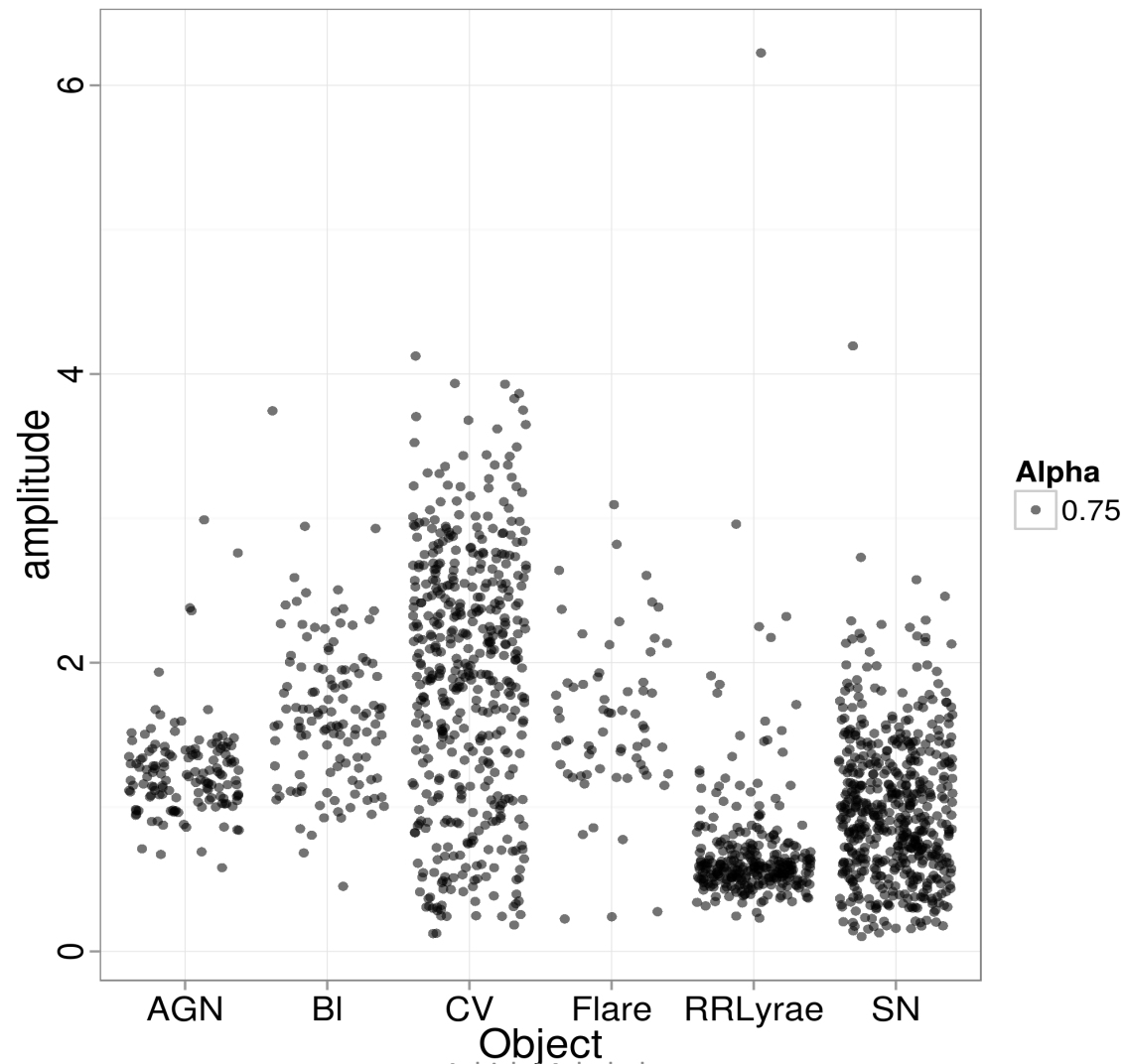


```
ggplot(data=crts_6class, aes(x=object, y=amplitude)) +  
geom_jitter()
```

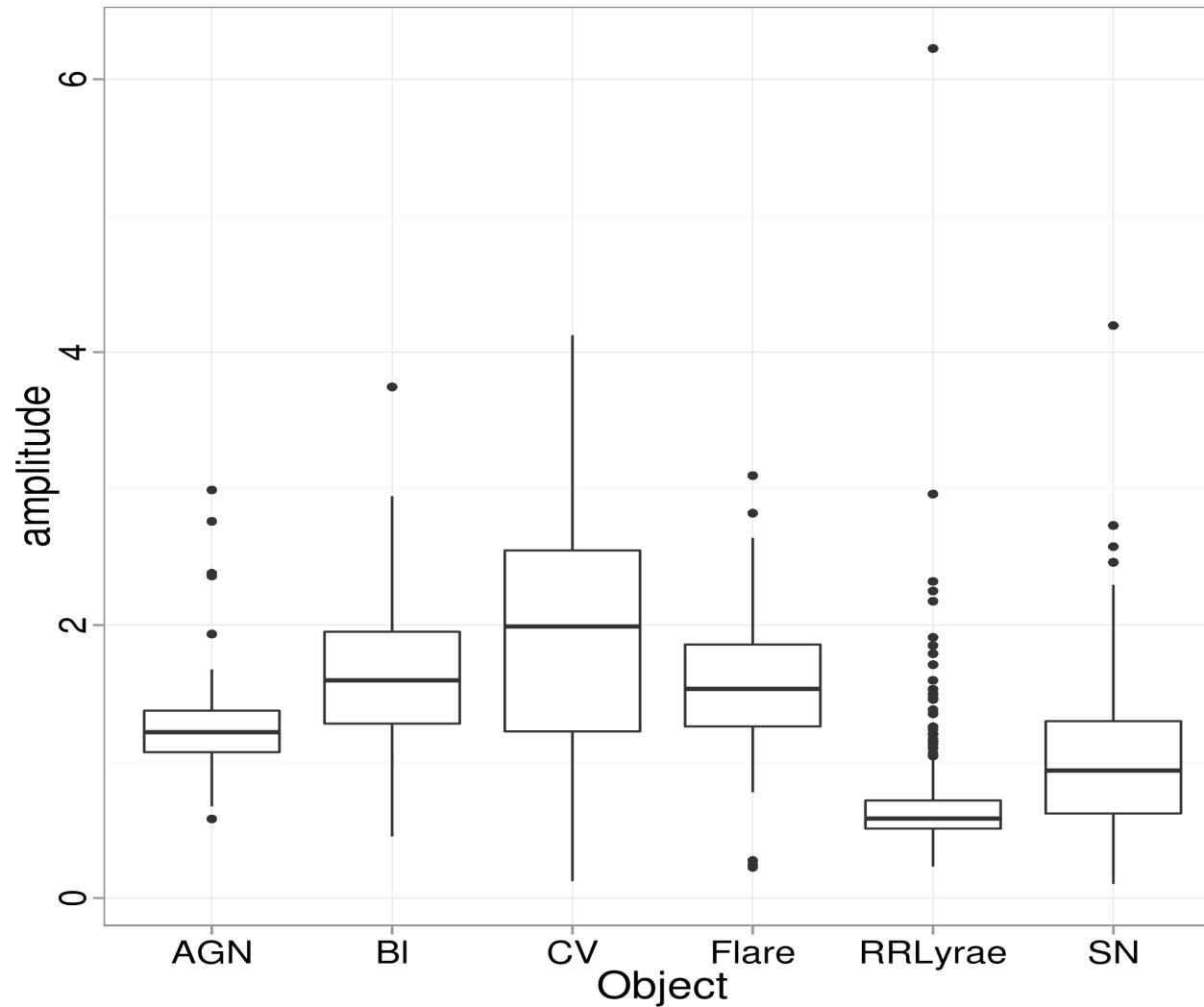


mitigates overplotting

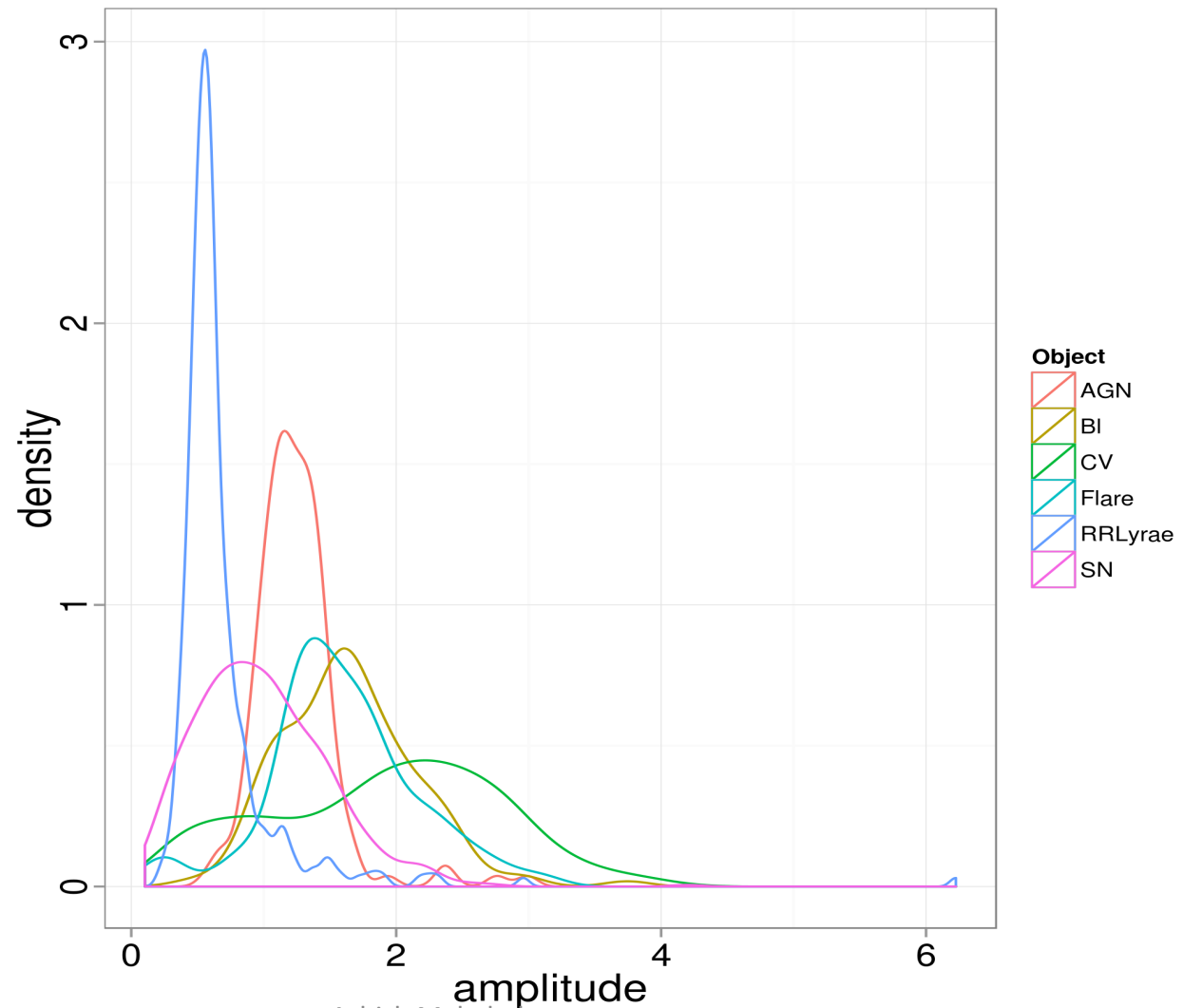

```
ggplot(data=crts_6class, aes(x=object, y=amplitude)) +  
geom_jitter(alpha=0.75)
```



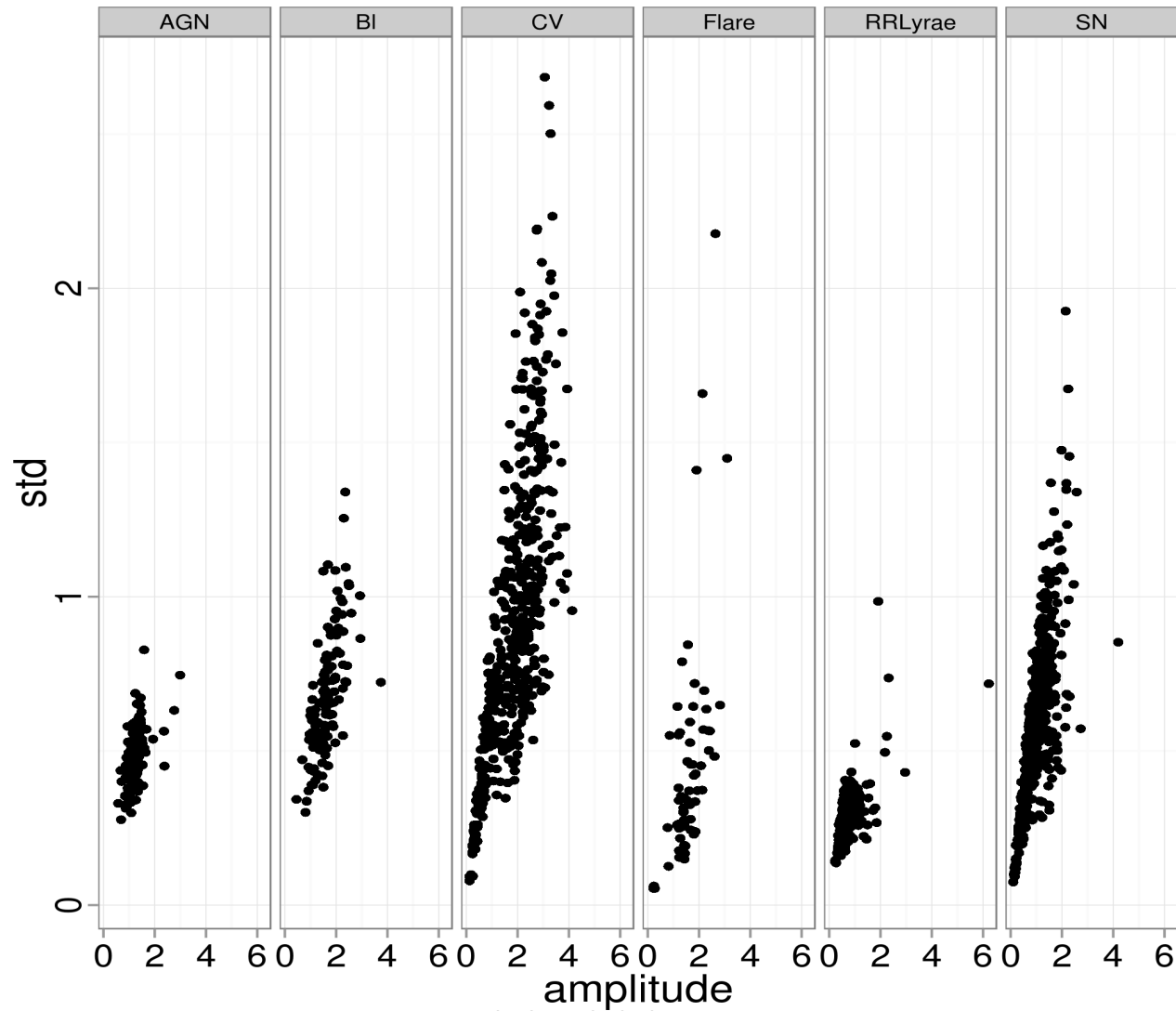
```
ggplot(data=crts_6class, aes(x=object, y=amplitude)) +  
geom_boxplot()
```



```
ggplot(data=crts_6class, aes(x=amplitude, colour=object)) +  
geom_density()
```



```
ggplot(data=crts_6class, aes(x=amplitude, y=std)) + geom_point() +  
facet_grid(~object)
```



```
ggplot(data=crts_6class, aes(x=amplitude, y=std)) + geom_point() +  
geom_smooth() + facet_grid(.~object)
```

