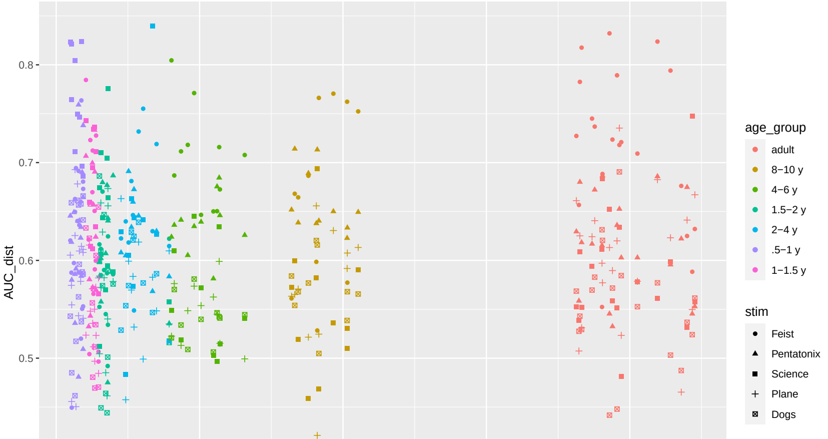
Mappings and datasets

John Franchak

Multiple mappings

The graphs we've made so far have only mapped x and y. The aes() commands in ggplot lets us map multiple aesthetics:

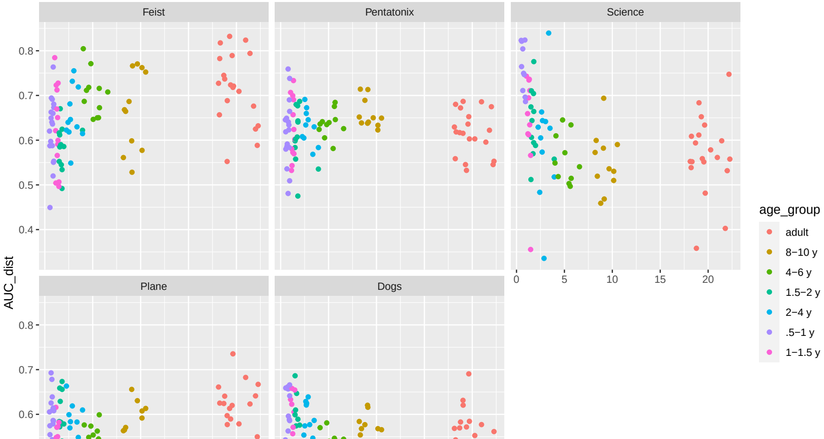
```
ds %>%
  ggplot(aes(x = age, y = AUC_dist, shape = stim, color = a
  geom_point()
```



Another one

Try swapping things between mapped aesthetics and facets to find your best arrangement:

```
ds %>% ggplot(aes(x = age, y = AUC_dist, color = age_group)
  geom_point() +
  facet_wrap("stim")
```



Using multiple data sets in a plot

Graphing individual data is better than just showing a bar, but shouldn't we have both summary and raw data? One way to do this is to create a summary data set and use two data sets mapped to different geoms. It looked too busy with a black bar over blue to black points, so I went with semi-transparent (use `alpha`) gray points with a black bar. `ggplot` calls layer geoms in order, so whatever you want in the foreground should be last in the call.

```r
ds_long_summary <- ds_long %>%
  group_by(model, stim) %>%
  summarize(mean_auc = mean(AUC, na.rm = T)) %>%
  ungroup()
```

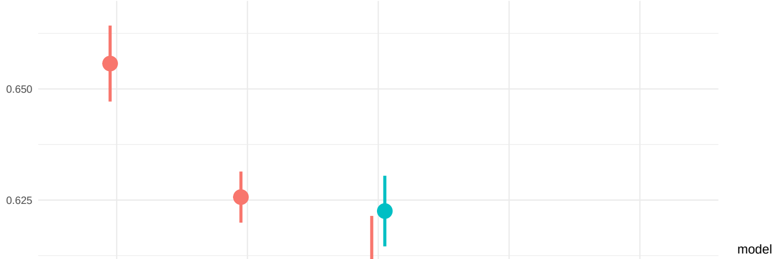
```
## `summarise()` has grouped output by 'model'. You can ove
```

```r
ggplot() +
  geom_sina(data = ds_long, aes(x = model, y = AUC), color
  geom_point(data = ds_long_summary, aes(x = model, y = mea
  facet_wrap("stim") + theme_minimal()
```

Using stat_summary

But this is a statistics programming language! Do we really *need* to calculate the upper and lower bounds of the error bars manually? No, but I think it's helpful to know how to manually map each part of a geom. Some solutions won't have a shortcut, so knowing exactly how to summarize your data or pull from multiple datasets helps you to understand what makes a graph. But if you want to write a cleaner mean and error bar summary, try `stat_summary`:

```
ds_long %>% drop_na(AUC) %>% ggplot(aes(x = stim, y = AUC,
  stat_summary(fun.data = mean_se, geom = "pointrange", siz
  theme_minimal()
```


model