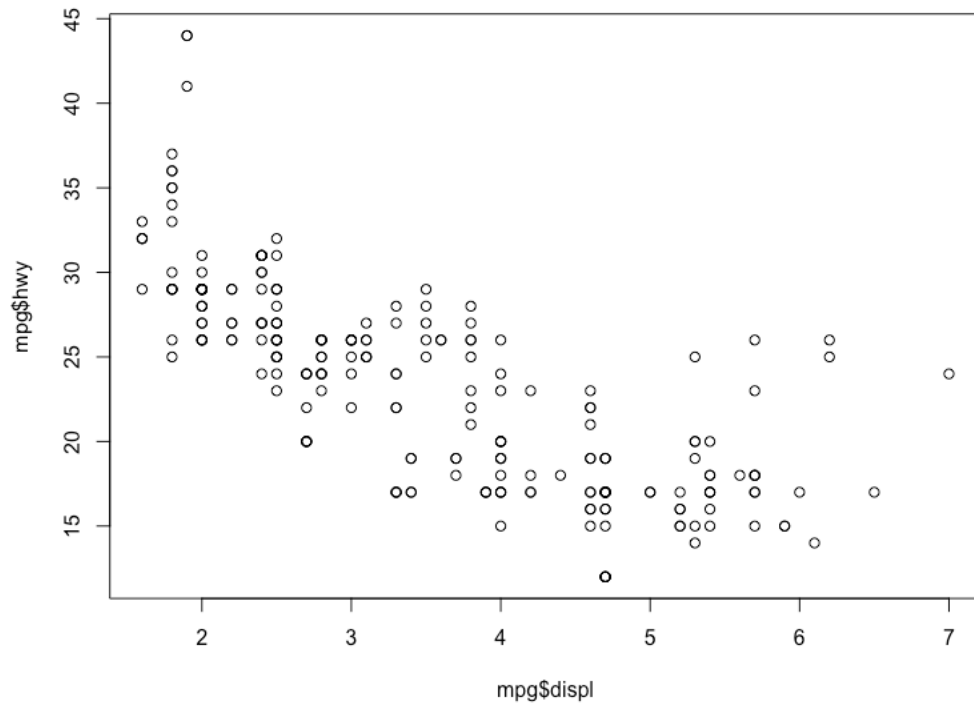


# The logic behind ggplot syntax

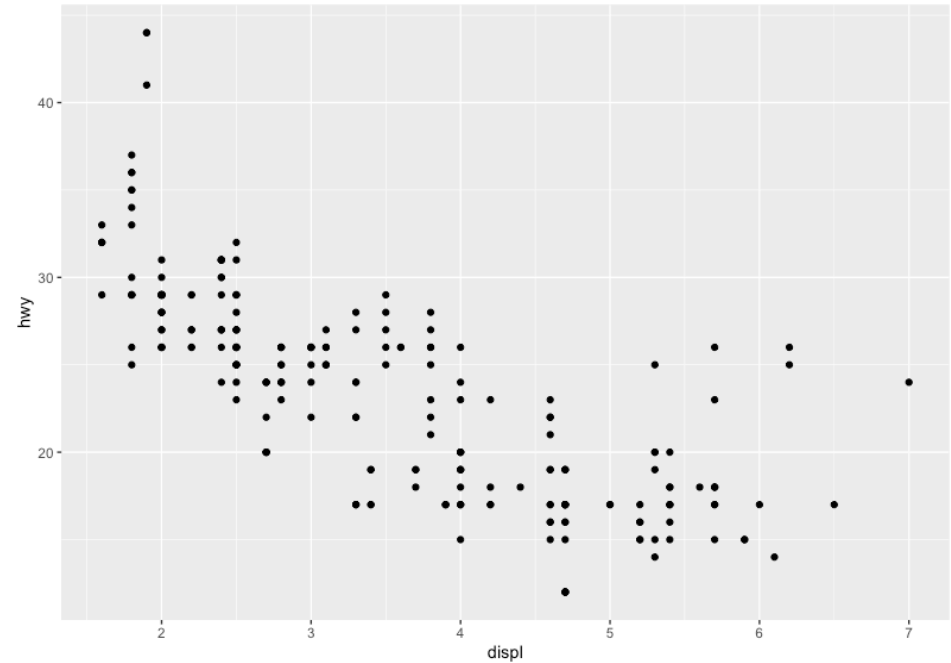
(Slides adapted from O. Shalem from GCB535)

I just got done telling you about plotting in base R (left).  
But you could have plotted this in ggplot2 (the right)!



Base plotting:

```
plot(mpg$displ,mpg$hwy)
```

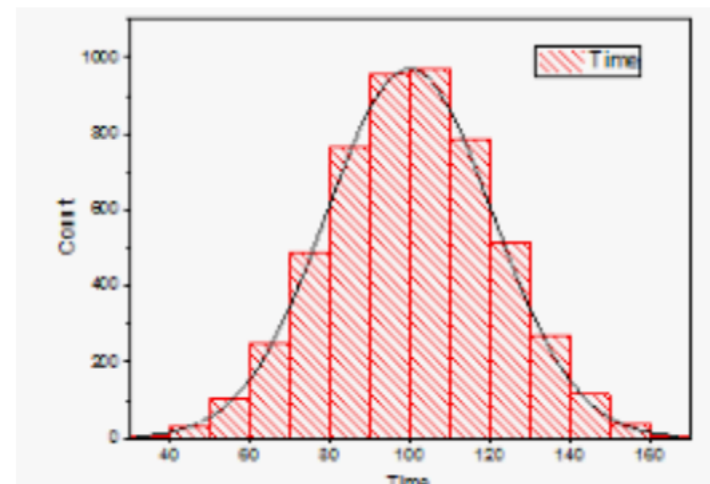
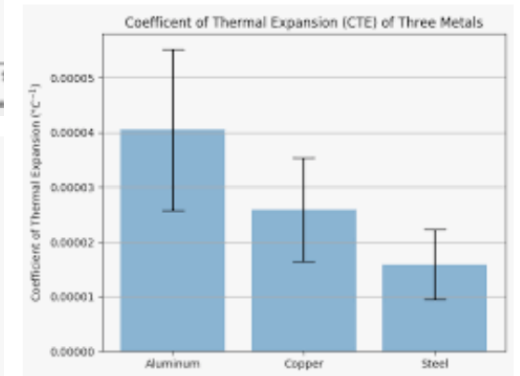
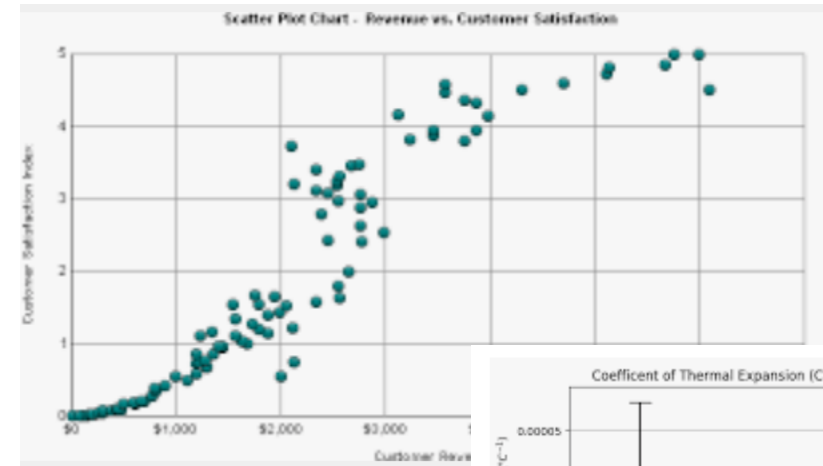
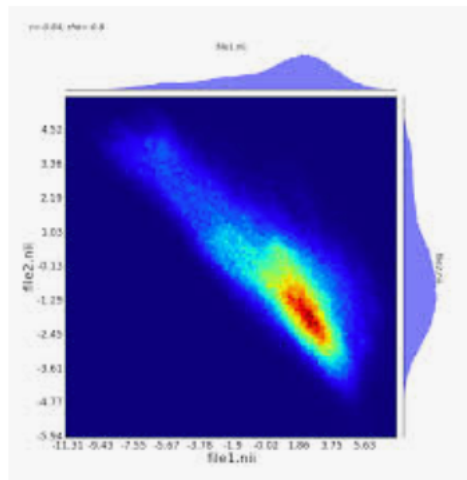
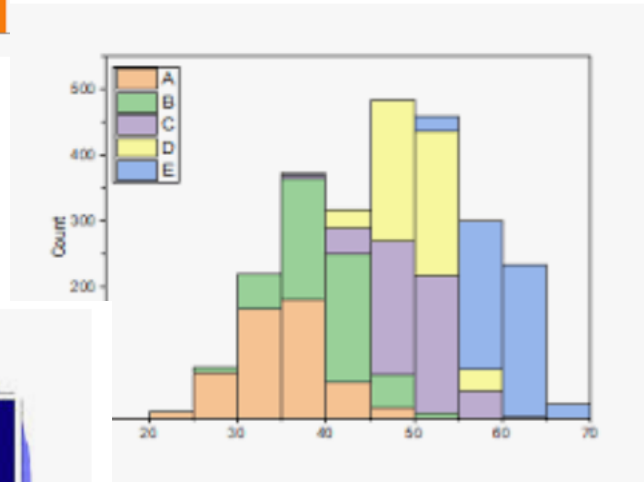
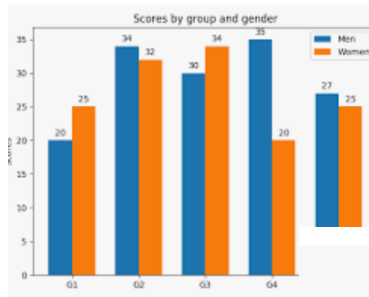


ggplot:

```
ggplot(data=mpg) + geom_point(mapping = aes(x=displ,y=hwy))
```

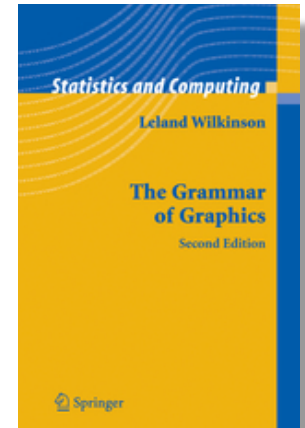
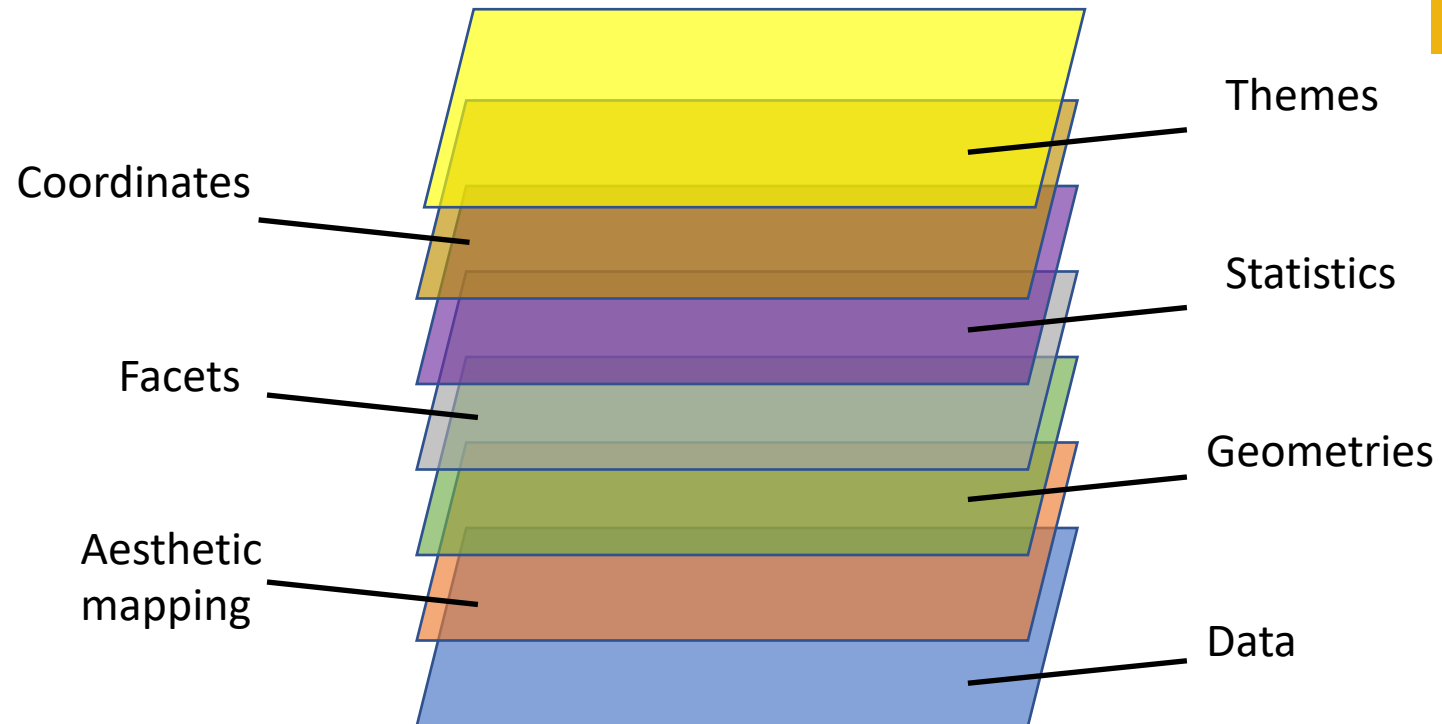
Why????

# Can we generate a single grammar that will be able to describe all types of scientific data graphics?



# ggplot stands for grammar of graphics plot

- Lealand Wilkinson, Grammar of Graphics 1999
- Implemented in R by Hadley Wickham
- Layered grammatical elements

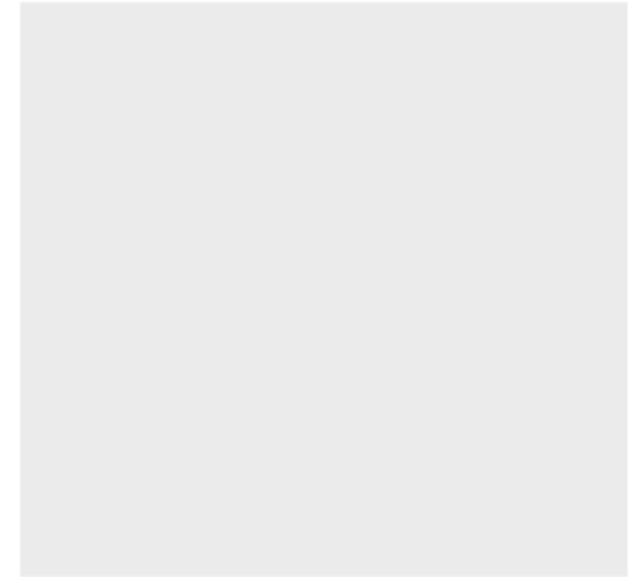


# The data and aesthetic layers

```
ggplot(data=mpg)
```

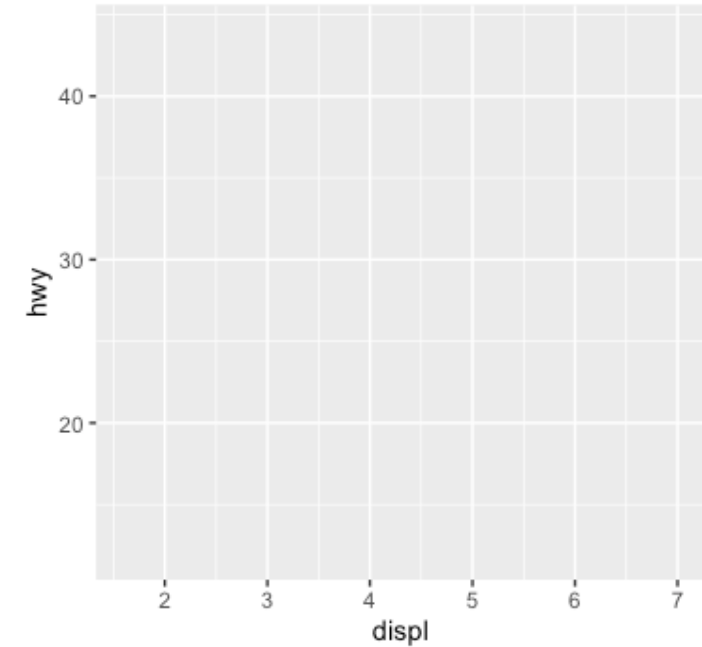
```
# A tibble: 6 x 11
```

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
	<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<int>	<int>	<chr>	<chr>
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact



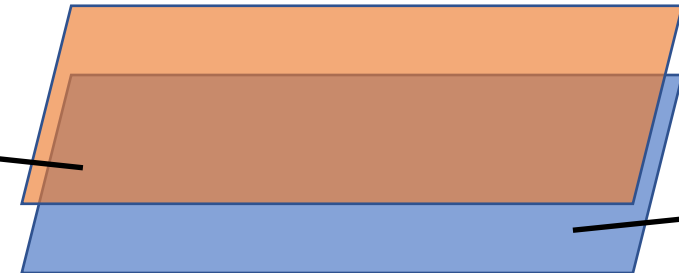
# The data and aesthetic layers

```
ggplot(data=mpg, mapping=aes(x=displ, y=hwy))
```



Aesthetic  
mapping

Data

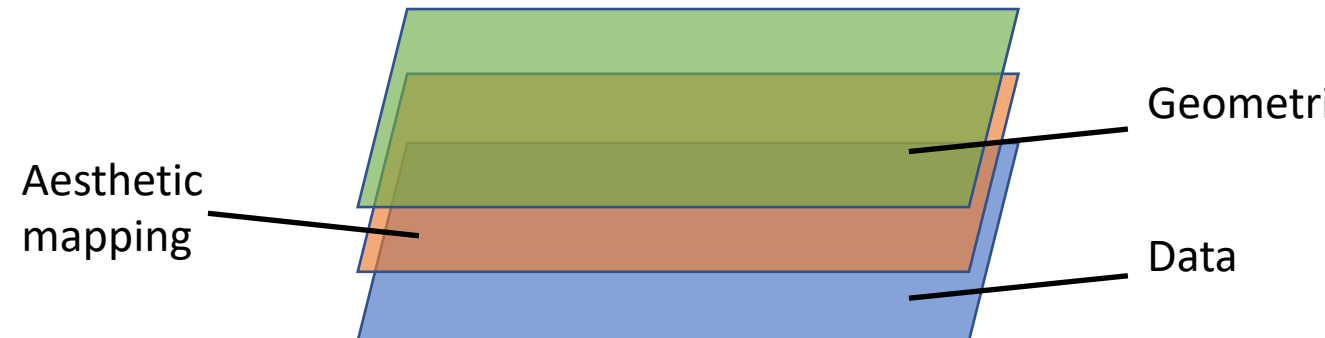
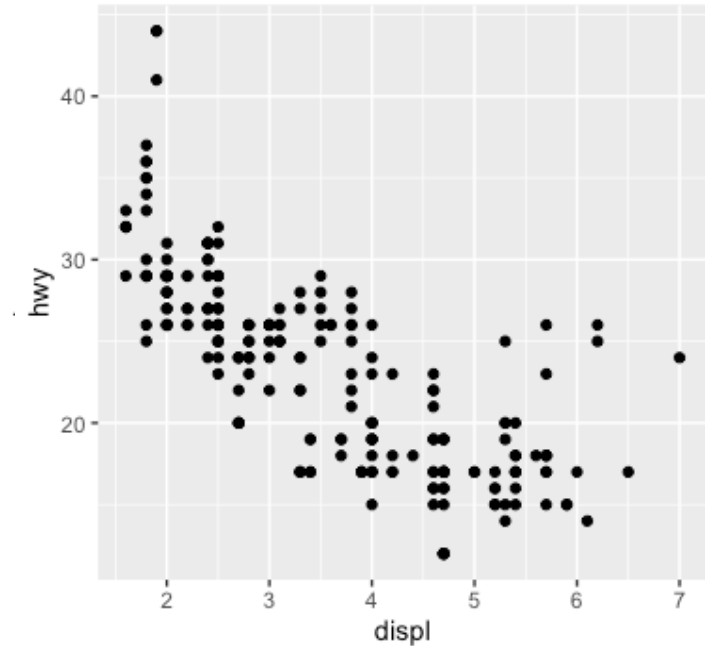


# geom layers

```
ggplot(data=mpg, mapping=aes(x=displ, y=hwy)) +  
geom_point()
```

```
> ls(pattern = '^geom_', env = as.environment('package:ggplot2'))
```

[1] "geom_abline"	"geom_area"	"geom_bar"	"geom_bin2d"	"geom_blank"
[6] "geom_boxplot"	"geom_col"	"geom_contour"	"geom_contour_filled"	"geom_count"
[11] "geom_crossbar"	"geom_curve"	"geom_density"	"geom_density_2d"	"geom_density_2d_filled"
[16] "geom_density2d"	"geom_density2d_filled"	"geom_dotplot"	"geom_errorbar"	"geom_errorbarh"
[21] "geom_freqpoly"	"geom_function"	"geom_hex"	"geom_histogram"	"geom_hline"
[26] "geom_jitter"	"geom_label"	"geom_line"	"geom_linerange"	"geom_map"
[31] "geom_path"	"geom_point"	"geom_pointrange"	"geom_polygon"	"geom_qq"
[36] "geom_qq_line"	"geom_quantile"	"geom_raster"	"geom_rect"	"geom_ribbon"
[41] "geom_rug"	"geom_segment"	"geom_sf"	"geom_sf_label"	"geom_sf_text"
[46] "geom_smooth"	"geom_spoke"	"geom_step"	"geom_text"	"geom_tile"
[51] "geom_violin"	"geom_vline"			

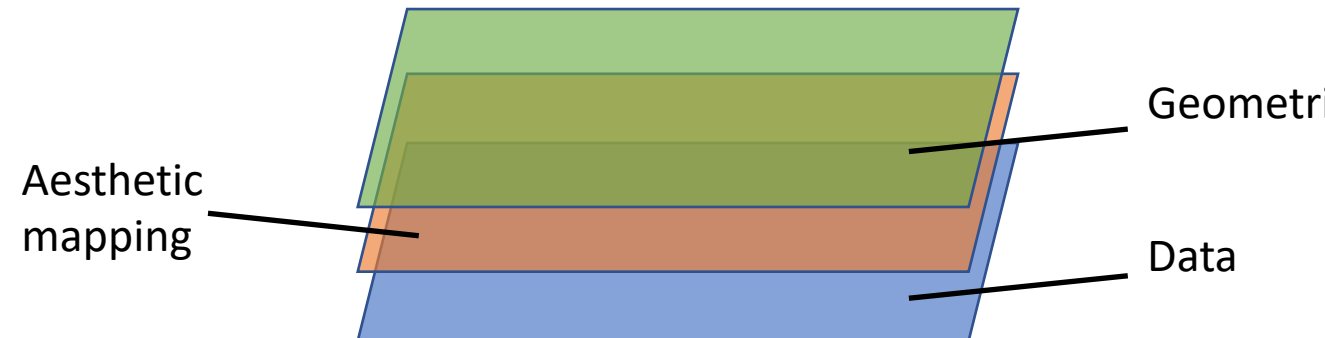
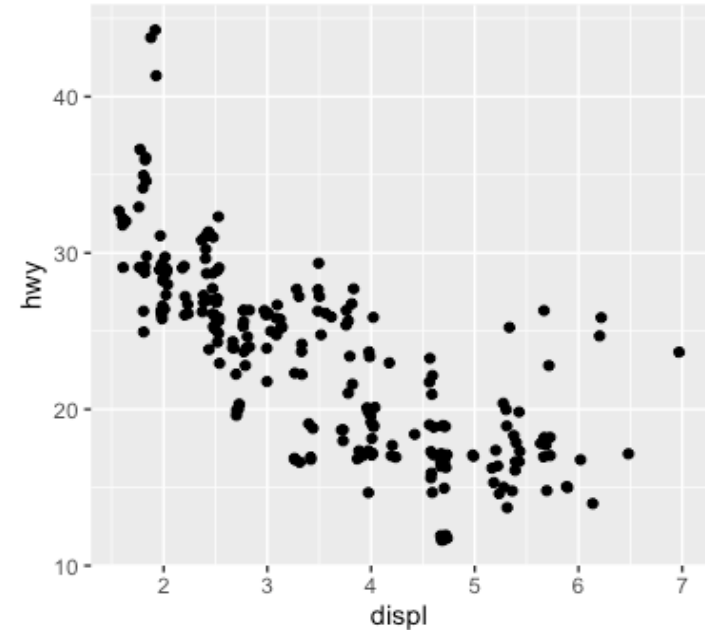


# We can easily add aesthetics and add/change geoms

```
ggplot(data=mpg, mapping=aes(x=displ, y=hwy)) +  
geom_jitter()
```

```
> ls(pattern = '^geom_', env = as.environment('package:ggplot2'))
```

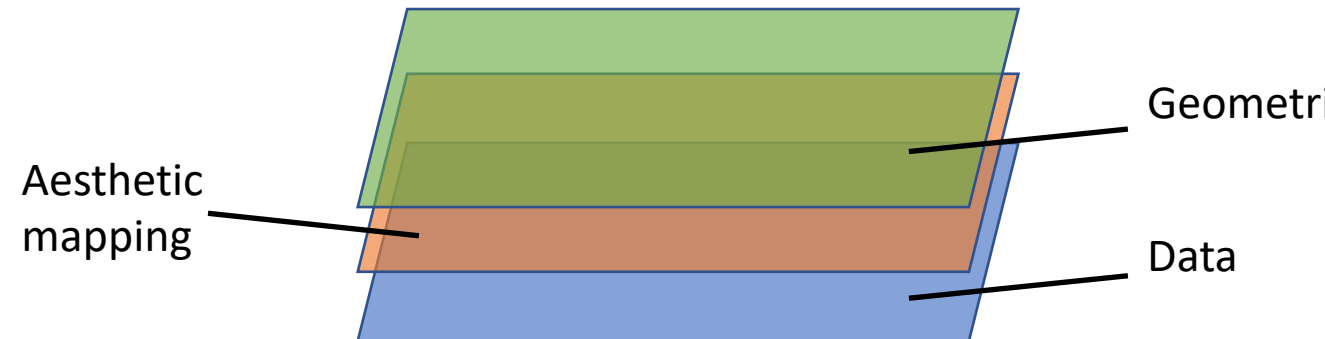
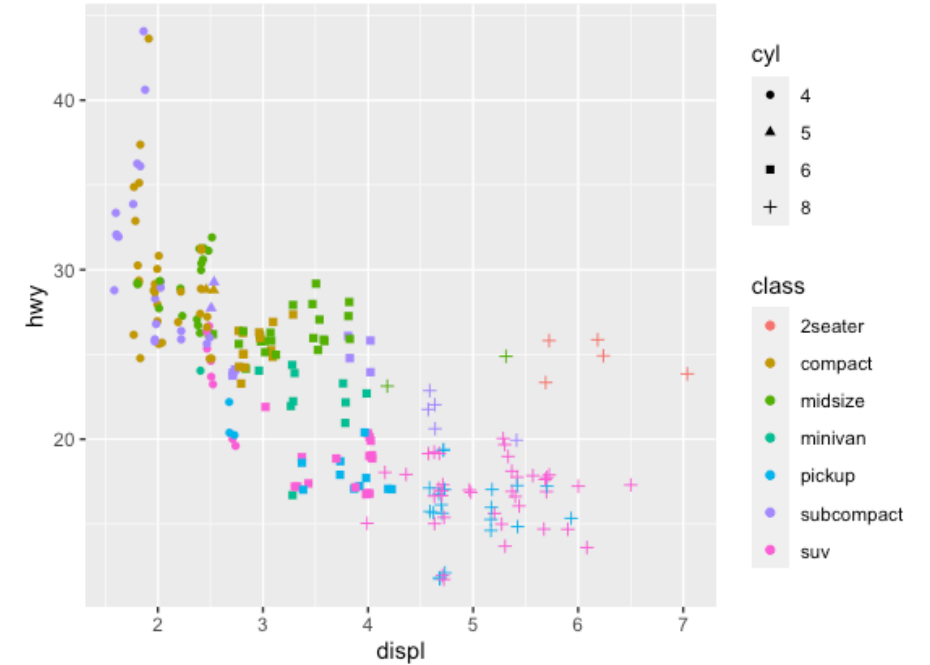
[1] "geom_abline"	"geom_area"	"geom_bar"	"geom_bin2d"	"geom_blank"
[6] "geom_boxplot"	"geom_col"	"geom_contour"	"geom_contour_filled"	"geom_count"
[11] "geom_crossbar"	"geom_curve"	"geom_density"	"geom_density_2d"	"geom_density_2d_filled"
[16] "geom_density2d"	"geom_density2d_filled"	"geom_dotplot"	"geom_errorbar"	"geom_errorbarh"
[21] "geom_freqpoly"	"geom_function"	"geom_hex"	"geom_histogram"	"geom_hline"
[26] "geom_jitter"	"geom_label"	"geom_line"	"geom_linerange"	"geom_map"
[31] "geom_path"	"geom_point"	"geom_pointrange"	"geom_polygon"	"geom_qq"
[36] "geom_qq_line"	"geom_quantile"	"geom_raster"	"geom_rect"	"geom_ribbon"
[41] "geom_rug"	"geom_segment"	"geom_sf"	"geom_sf_label"	"geom_sf_text"
[46] "geom_smooth"	"geom_spoke"	"geom_step"	"geom_text"	"geom_tile"
[51] "geom_violin"	"geom_vline"			





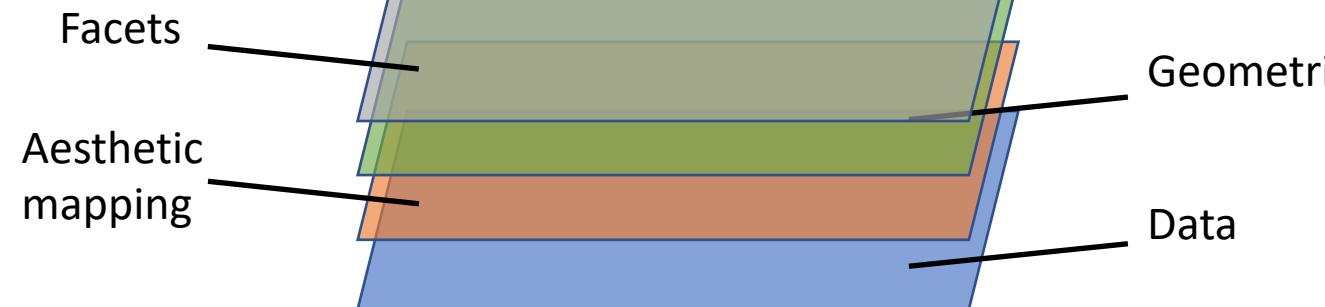
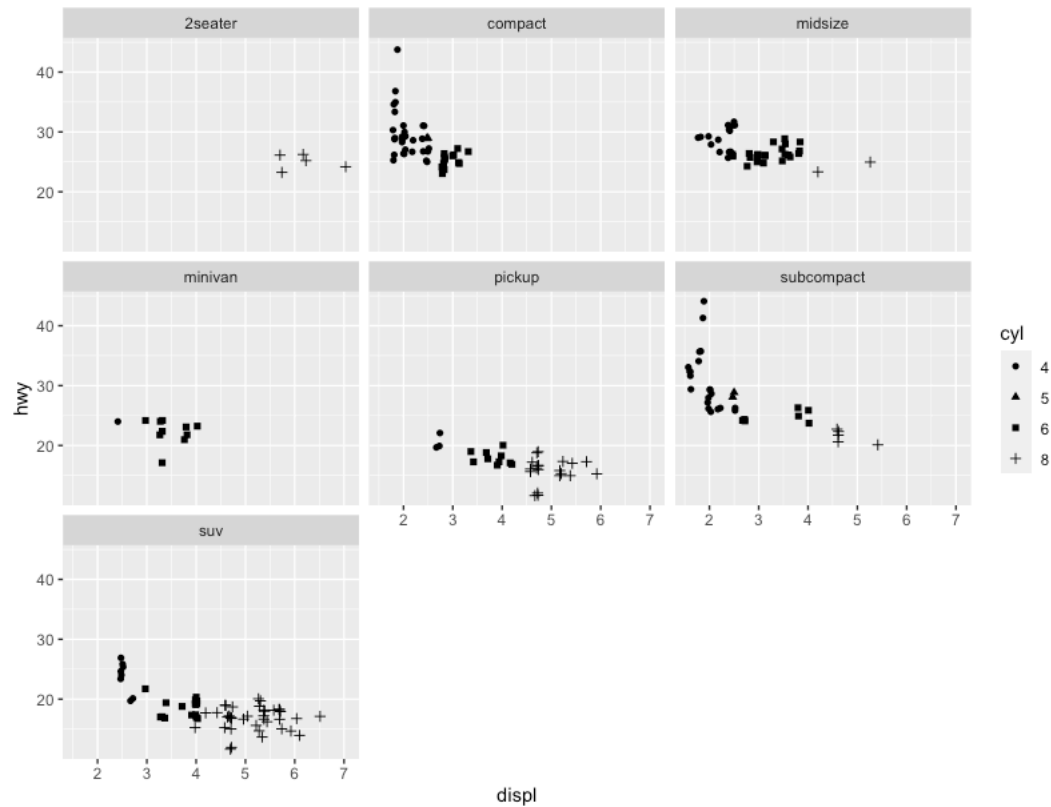
# We can easily add aesthetics and add/change geoms

```
ggplot(data=mpg,mapping=aes(x=displ,y=hwy, ...  
col=class,shape=cyl)) + geom_jitter()
```



# Facets are used to divide the data to groups

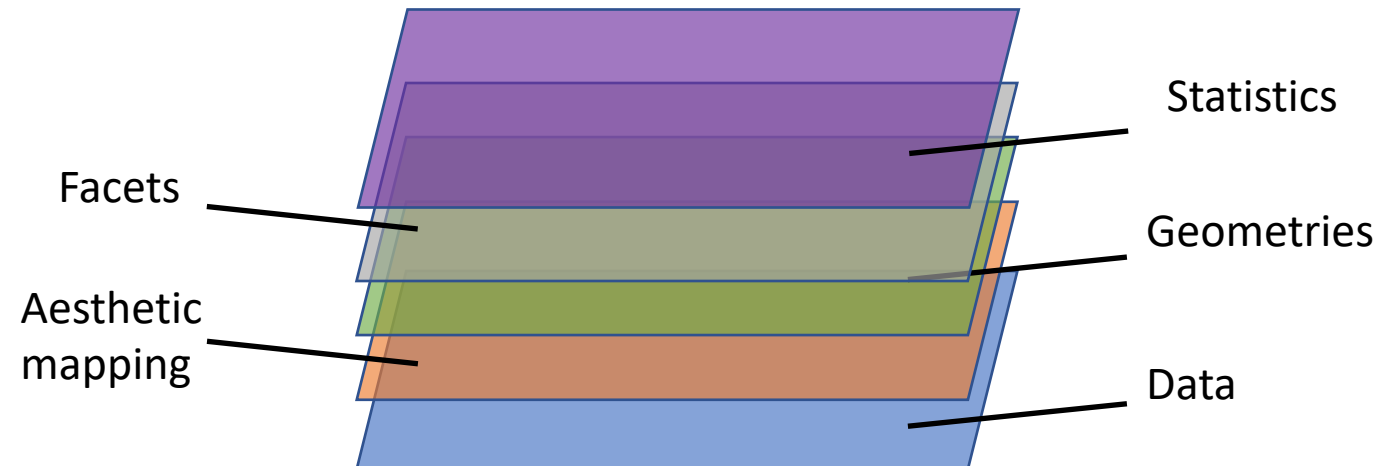
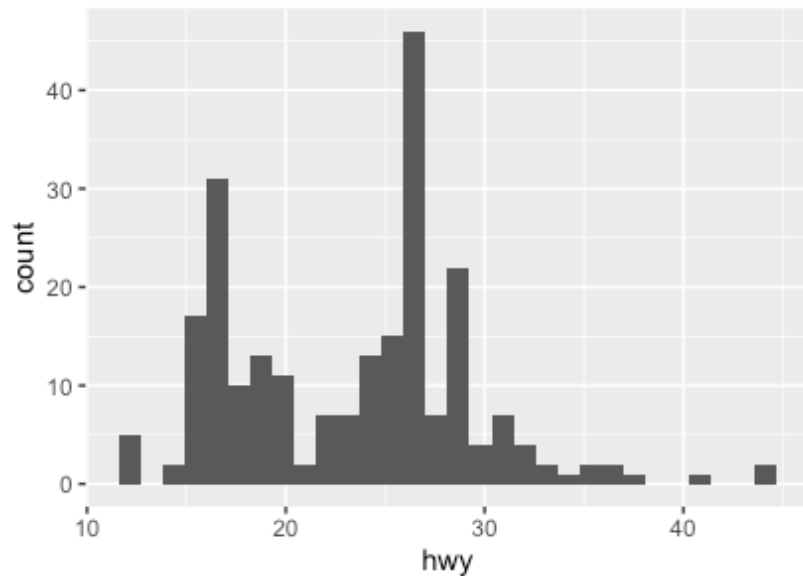
```
ggplot(data=mpg,mapping=aes(x=displ,y=hwy, ...  
shape=cyl)) + geom_jitter() +  
Facet_wrap(~class)
```



# Stats are values that are calculated from the data and mapped on aesthetics

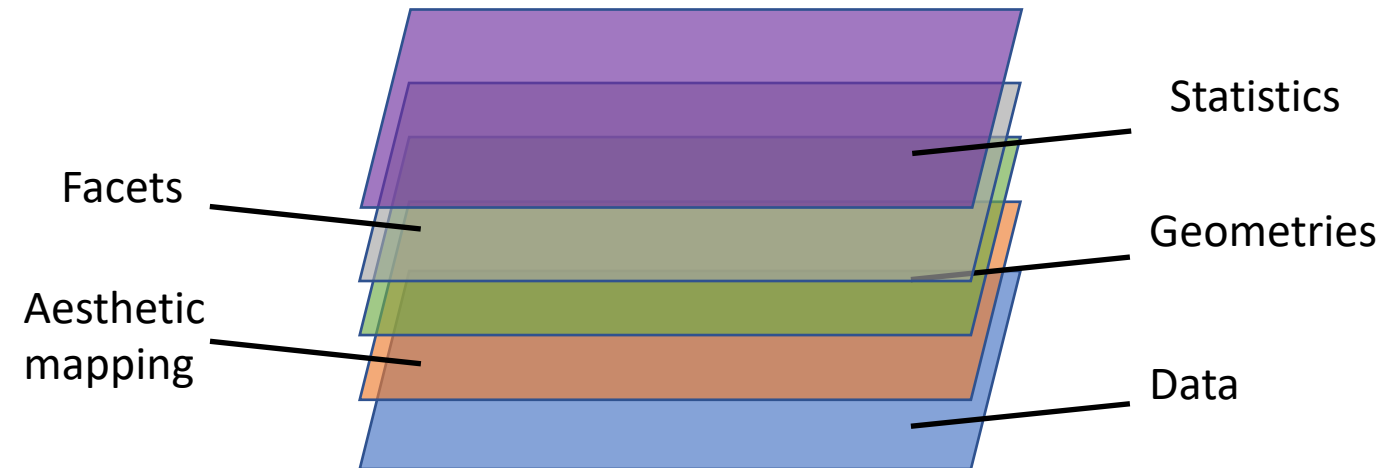
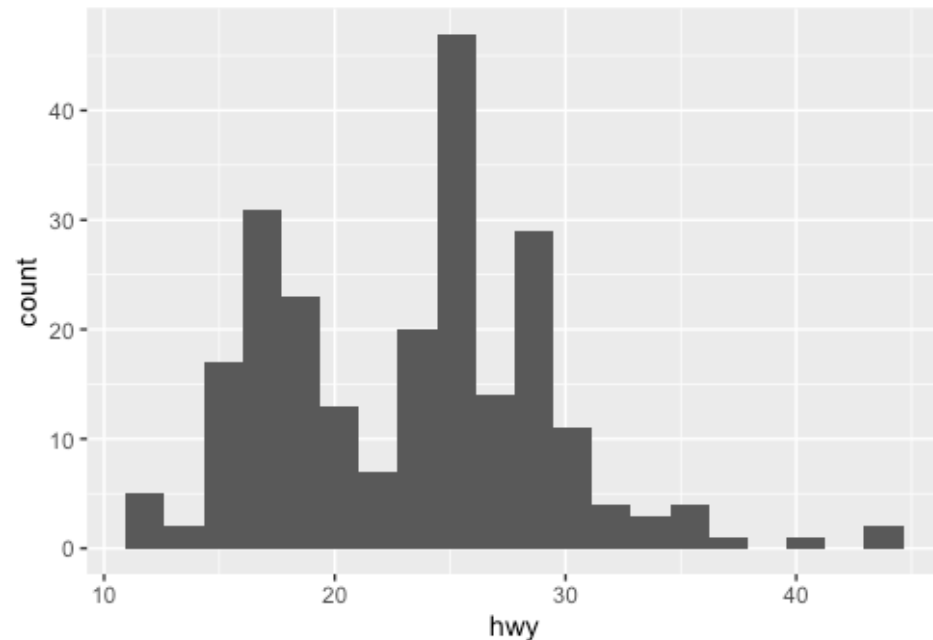
```
ggplot(data=mpg,mapping=aes(x=hwy)) + geom_histogram()
```

```
> ggplot(data=mpg,mapping=aes(x=hwy)) + geom_histogram()  
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



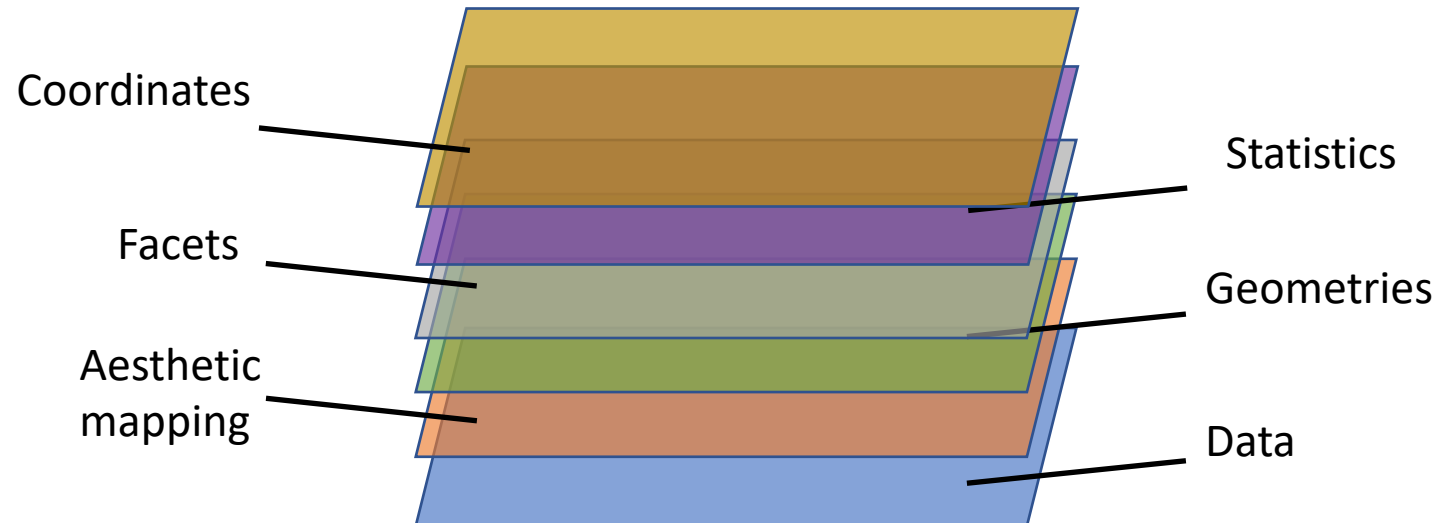
Stats are values that are calculated from the data and mapped on aesthetics

```
ggplot(data=mpg,mapping=aes(x=hwy)) + geom_histogram(stat='bin',bins = 20)
```



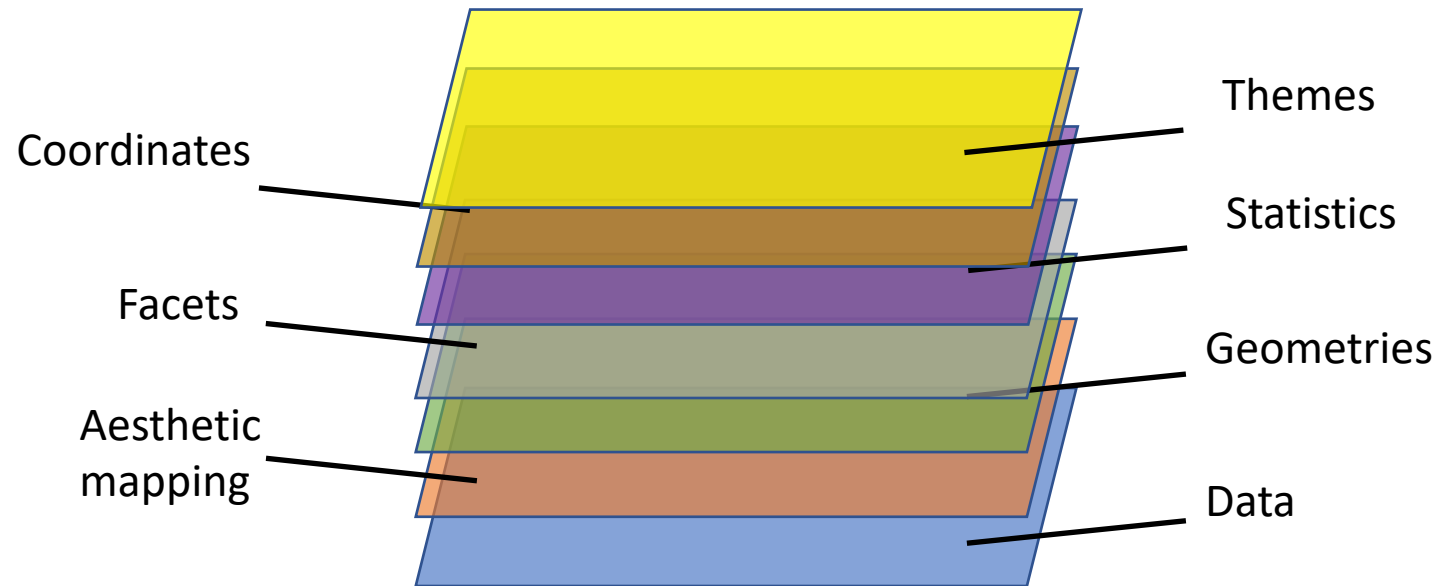
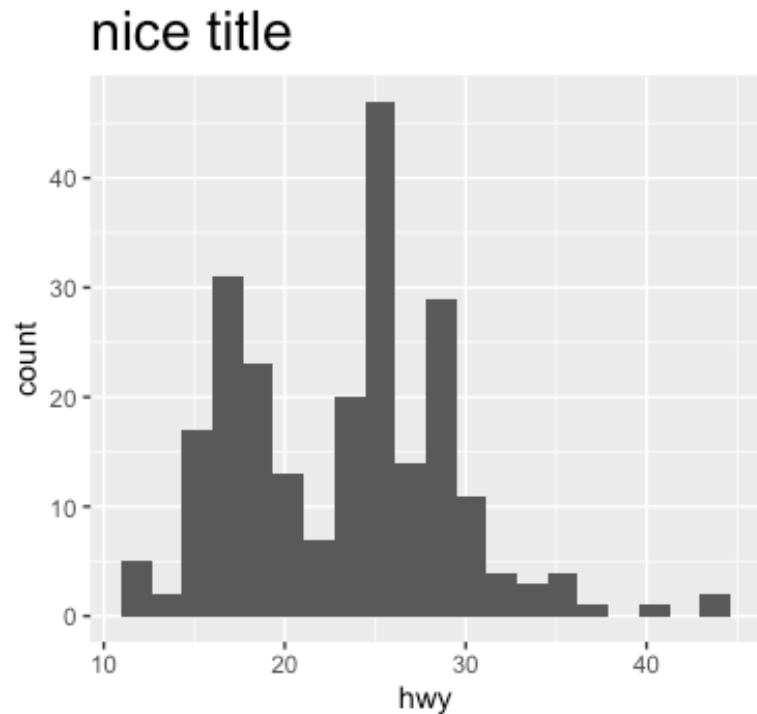
# Coordinates defines the actual position in 2d space for the aesthetic mappings

```
coord_cartesian()  
Coord_polar()  
coord_trans(y = "log10")  
scale_x_continuous(limits = c(4, 6)) / xlim(4,6)  
.  
.  
.  
.
```



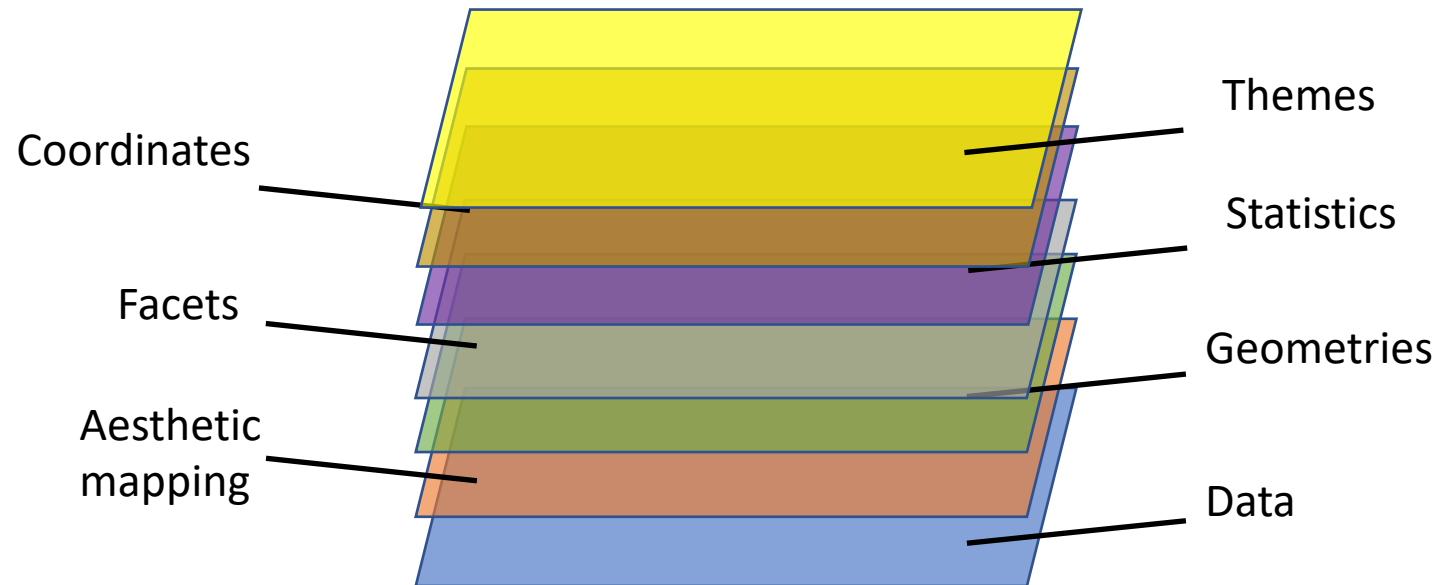
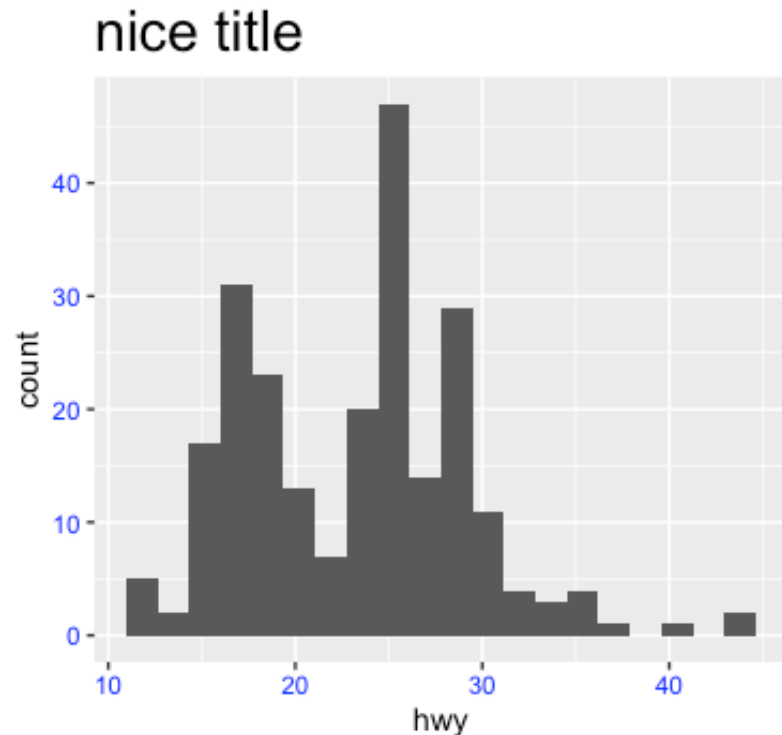
# Themes help define all the graphics that are not related to the data mappings

```
ggplot(data=mpg,mapping=aes(x=hwy)) + geom_histogram(stat='bin',bins = 20) +  
labs(title='nice title') + theme(plot.title=element_text(size=20) + )
```



# Themes help define all the graphics that are not related to the data mappings

```
ggplot(data=mpg,mapping=aes(x=hwy)) + geom_histogram(stat='bin',bins = 20) +  
labs(title='nice title') + theme(plot.title=element_text(size=20) + axis.text =  
element_text(colour = "blue"))
```



# Themes help define all the graphics that are not related to the data mappings

```
ggplot(data=mpg,mapping=aes(x=hwy)) + geom_histogram(stat='bin',bins = 20) +  
labs(title='nice title') + theme_classic()
```

