

ggplot2

Much funnier if you speak Mandarin

Samuel Robinson, Ph.D.

October 29, 2020

Motivation

What is ggplot2?

- ggplot philosophy
- Simple plots
- Some useful techniques
- More complicated plots

What is ggplot2?

- Updated version of `ggplot` (older R package)
- Implementation of Wilkinson's *grammar of graphics*
- Elements: data, transformations, elements, scale, guide, coordinates
- Describes a **layered approach to building graphics** beyond formulaic plots (e.g. "boxplot", "scatterplot")
- Many different extensions available [here](#)

Philosophy:

- Data input centered around `data.frames`
- Data display centered around `geoms` (geometric objects)
- Columns from data frames are mapped into `geoms` using `aesthetics`
- `geoms` are displayed according to `themes`

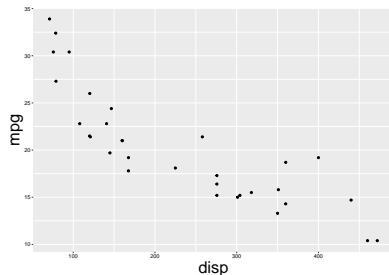
Simple example - scatterplot

```
data(mtcars) # mtcars dataset (built into R)
head(mtcars,5) # Show first 5 rows
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
## Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

```
# Load ggplot library
library(ggplot2)

# Top line of code says:
# - data from mtcars dataframe
# - aes = aesthetics from dataframe
# - map disp to x-axis, mpg to y-axis
ggplot(data = mtcars, aes(x = disp, y = mpg))+
  geom_point() # Display data using points
```



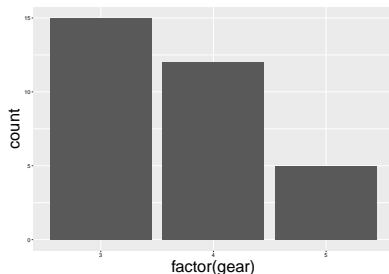
Simple example - bar plot

```
data(mtcars) # mtcars dataset (built into R)
head(mtcars,5) # Show first 5 rows
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
## Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

```
# Top line of code says:
# - map gear (number of gears) to x-axis
#   - first converted to a factor
ggplot(data = mtcars, aes(x = factor(gear)))+
  geom_bar()
# Display number of data points for each factor

# Automatically uses stat='count' to group
# data according to factor
```

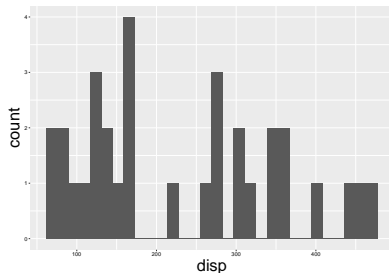


Simple example - histogram

```
data(mtcars) # mtcars dataset (built into R)
head(mtcars,5) # Show first 5 rows
```

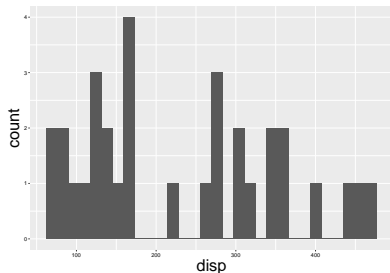
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
## Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2

```
# Top line of code says:
# - map disp (displacement) to x-axis
ggplot(data = mtcars, aes(x = disp))+
  # Group disp into bins, and display
  # count in each bin
  geom_histogram()
```



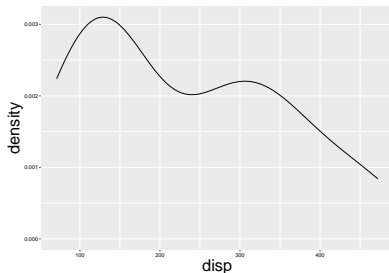
Simple example - histograms and density plots

```
# Histogram  
ggplot(data=mtcars, aes(x=disp)) +  
  geom_histogram()
```



Histogram

```
# Density plot  
ggplot(data=mtcars, aes(x=disp)) +  
  geom_density()
```



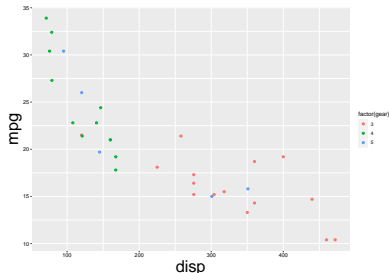
Probability density plot

$$\int_{-\infty}^{\infty} f(x) dx = 1$$

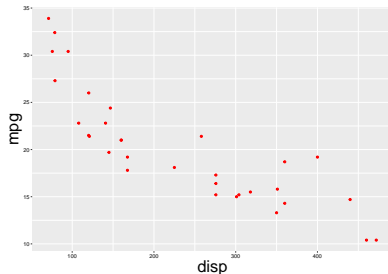
Colours in plots

- Colours can be *mapped* (via aes) or *set* (outside of aes)

```
ggplot(data=mtcars, aes(x=displacement, y=mpg)) +  
  # Maps gear to colour  
  geom_point(aes(col=factor(gear)))
```



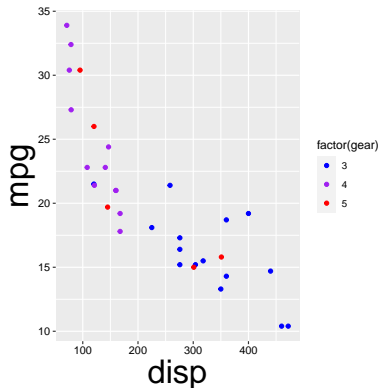
```
ggplot(data=mtcars, aes(x=displacement, y=mpg)) +  
  geom_point(colour='red') #Sets colour
```



- Notice how aes was used twice in Figure 1? If used within the ggplot command, the rest of the geoms will remember it. Used within a geom, it will *update* the aesthetic

What if I want different colours?

- Default colour themes are pretty bad. Change them with `scale_colour_manual`
- Use `scale_fill_manual` for area-based colours (e.g. bar plots, polygons)
- Remember, 10% of males are red-green colourblind!

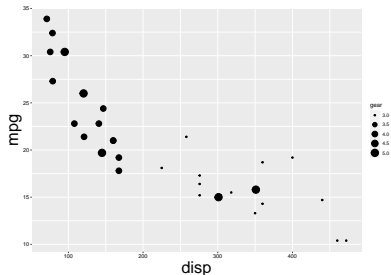


```
ggplot(data=mtcars, aes(x=disp, y=mpg)) +  
  geom_point(aes(col=factor(gear))) +  
  scale_colour_manual(values=c('blue', 'purple', 'red'))
```

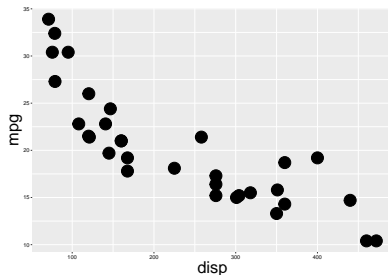
Sizes in plots

- Sizes can also be *mapped* (via aes) or *set* (outside of aes)

```
ggplot(data=mtcars, aes(x=disp, y=mpg)) +  
  # Maps gear to size  
  geom_point(aes(size=gear))
```



```
ggplot(data=mtcars, aes(x=disp, y=mpg)) +  
  geom_point(size=10) # Sets size
```

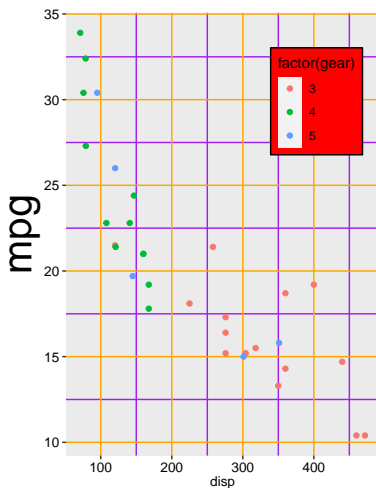


- Similar to colour choices, you can alter mapped sizes using `scale_size`

Change plot theme

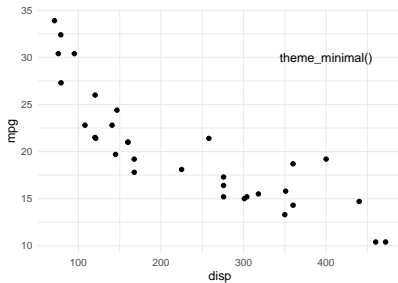
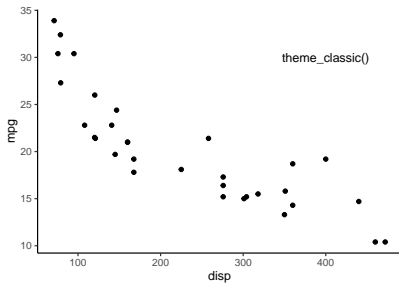
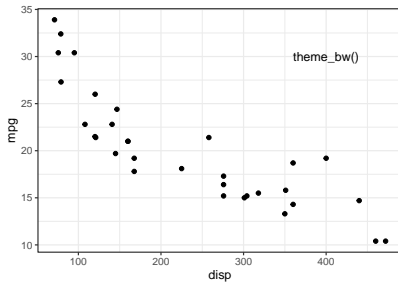
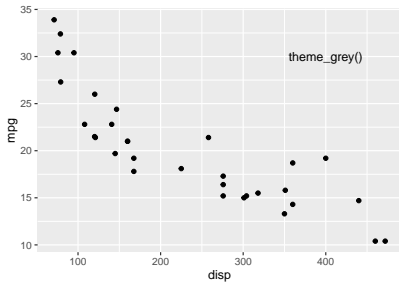
- theme controls almost all non-data elements of plots
- Made up of *elements*:
element_line(),
element_text(),
element_rect()
- Let's make some changes:

```
ggplot(data=mtcars, aes(x=disp, y=mpg)) +  
  # Maps gear to colour  
  geom_point(aes(col=factor(gear))) +  
  #Changes plot theme  
  theme(axis.title.x=element_text(size=10),  
        legend.background=element_rect(fill='red'),  
        legend.position=c(0.8,0.8),  
        panel.grid.minor=element_line(colour='purple'),  
        panel.grid.major=element_line(colour='orange'))
```



- This plot is hideous, but it gives you the idea!
- Use ?theme to see all options

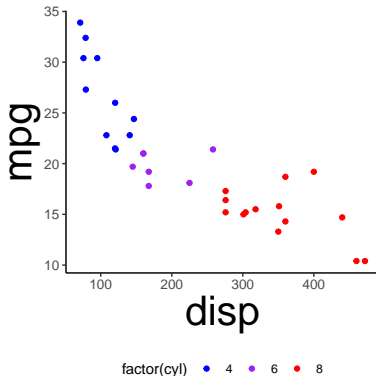
Preset themes



Make your own themes!

- You can modify existing themes in order to create your own
- Try using `theme_set()` at the start of your script to pre-set the theme for the rest of the script

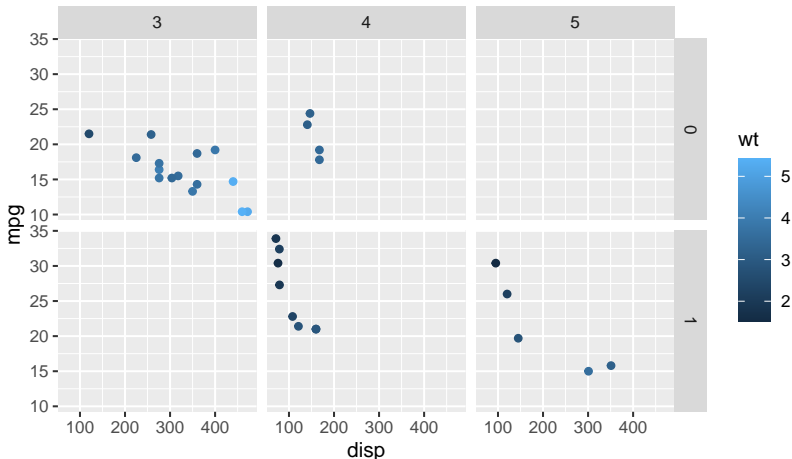
```
myTheme <- theme_classic()+ #Existing theme  
#Makes axis text bigger  
theme(axis.title=element_text(size=30),  
      axis.text=element_text(size=10),  
      legend.position='bottom')  
#Sets up this theme as "default"  
theme_set(myTheme)
```



Complex plots - facets

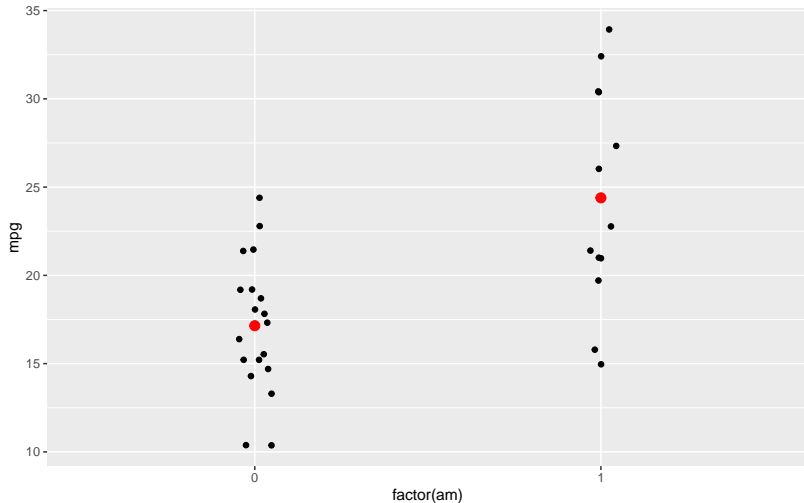
- It is possible to break up the plot into smaller facets that are mapped to a given variable
- This can be combined with colour/size mappings

```
ggplot(mtcars, aes(x=disp, y=mpg)) + geom_point(aes(col=wt)) +  
  facet_grid(factor(am) ~ factor(gear))
```



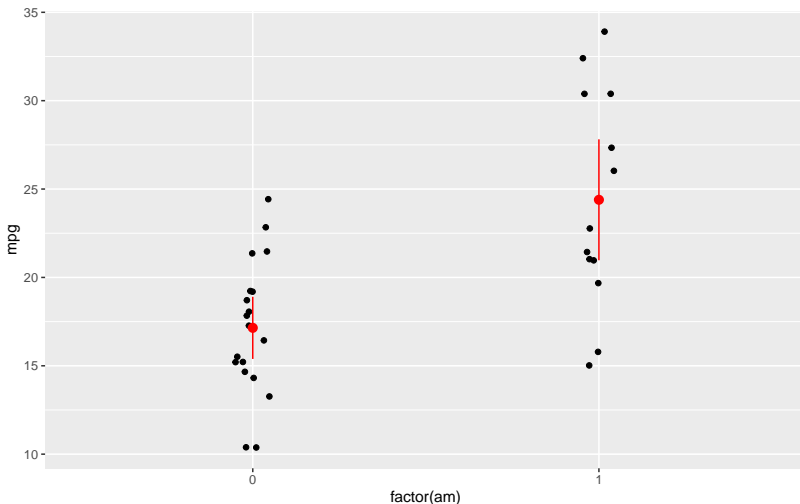
Complex plots - summary statistics (mean)

```
ggplot(mtcars, aes(x=factor(am), y=mpg)) +  
  geom_point(position=position_jitter(width=0.05)) + #Adds noise to data in x-dimension  
  geom_point(stat='summary', fun=mean, col='red', size=3) #Mean only
```



Complex plots - summary statistics (mean + SD)

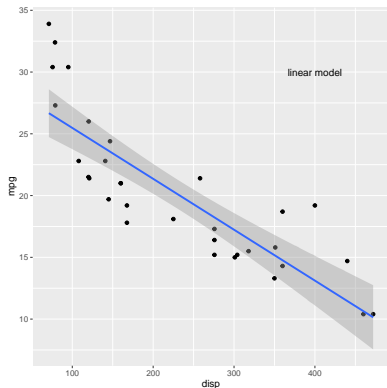
```
ggplot(arrange(mtcars, am, disp), aes(x=factor(am), y=mpg)) +  
  geom_point(position=position_jitter(width=0.05)) +  
  geom_pointrange(stat='summary', fun.data=mean_se,  
                 fun.args = list(mult = 2), col='red') #Mean + 2 SE
```



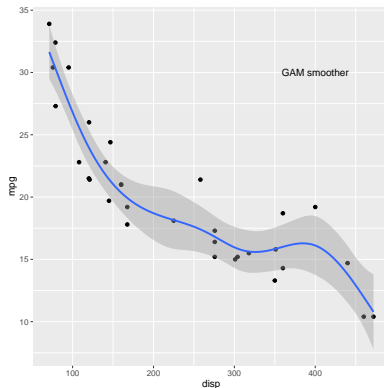
Complex plots - smoothers

- You can add `lm` (or other model) predictions to your plots:

```
ggplot(mtcars, aes(x=dis, y=mpg)) +  
  geom_point() +  
  geom_smooth(method='lm', formula=y~x)
```



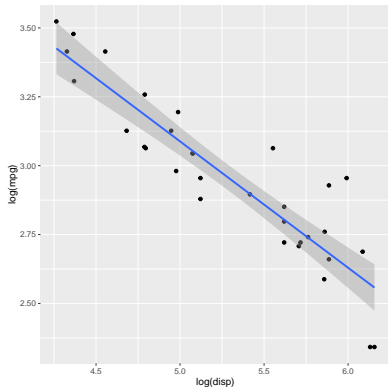
```
ggplot(mtcars, aes(x=dis, y=mpg)) +  
  geom_point() +  
  geom_smooth(method='gam', formula=y~s(x))
```



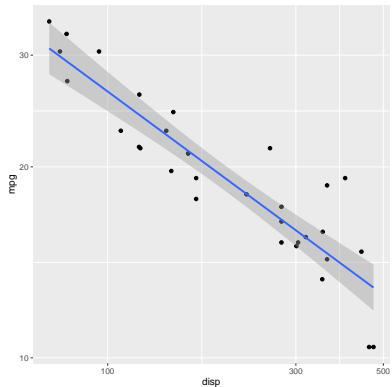
Complex plots - transformations

- You can show transformed data OR you can transform the axes themselves using `scale*_log10` (x or y axis)

```
ggplot(mtcars,aes(x=log(displ),y=log(mpg)))+  
  geom_point() +  
  geom_smooth(method='lm',formula=y~x)  
# Harder to interpret, because people can't  
# usually do log(x) in their head
```

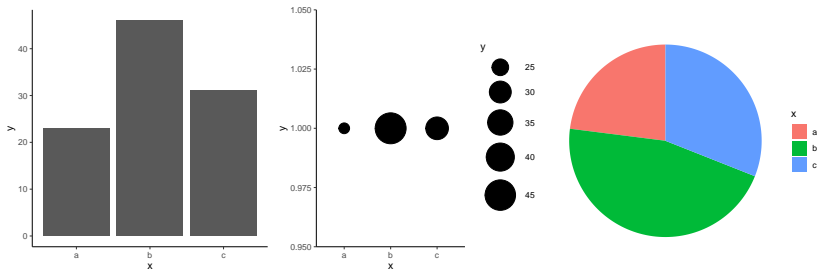


```
ggplot(mtcars,aes(x=displ,y=mpg))+  
  geom_point() +  
  geom_smooth(method='lm',formula=y~x)+  
  scale_x_log10() + scale_y_log10()  
# sqrt is also popular
```



Things to remember:

- Simpler plots are often better. Try to keep it to 3 aesthetics per panel. Avoid 3D plots.
- Making plots is iterative. Make a simple one and tweak it to improve it.
- Avoid “non-data ink” (see [Edward Tufte's](#) work)
- Our eyes are good at estimating linear positions, but bad at estimating area, volume, colour shading, and angles:



A challenger approaches:

Make these figures! Datasets are found in `mpg`, `msleep`, `trees`, and `starwars` (built into the `ggplot2` and `dplyr` packages)

