Introduction to ggplot

36-600

Fall 2021

ggplot

`ggplot` (actually, and perhaps confusingly, `ggplot2`) is "a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data [frame], tell `ggplot2` how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details."

Sounds good. Let's dive in:

```
suppressMessages(library(tidyverse))
```

ggplot: Basic Structure

A very basic call to `ggplot()` has the following structure:

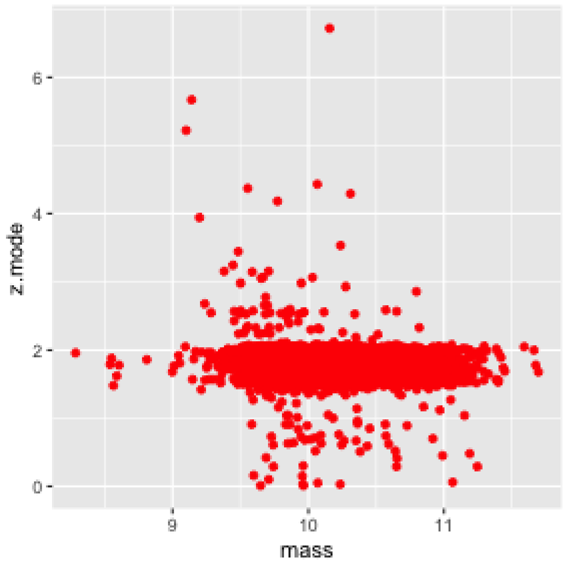
```
ggplot(data=<data frame>,mapping=aes(x=<x axis variable>,...)) + geom_<plot type>(<arguments>)
```

Let's read in the data frame that we use in the `dplyr` notes set:

```
df <- read.csv("http://www.stat.cmu.edu/~pfreeman/GalaxyMass.csv",stringsAsFactors=TRUE)
```

To plot, e.g., `z.mode` vs. `mass` (and remember: we plot y vs. x):

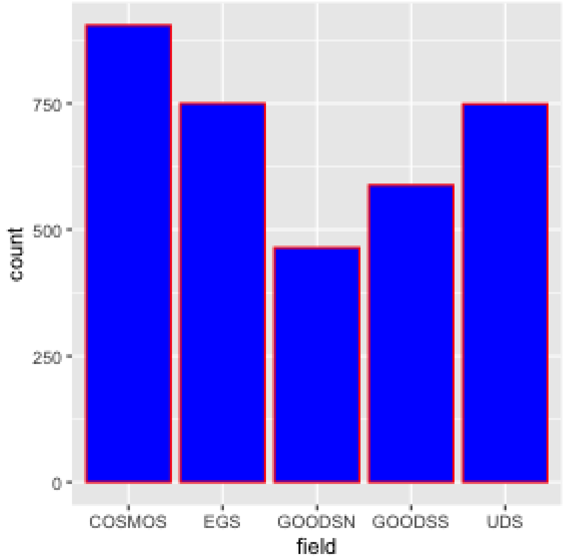
```
ggplot(data=df,mapping=aes(x=mass,y=z.mode)) +
  geom_point(color="red")
```



ggplot: Bar Chart

How many galaxies are in each field?

```
ggplot(data=df,mapping=aes(x=field)) +
  geom_bar(color="red",fill="blue")
```

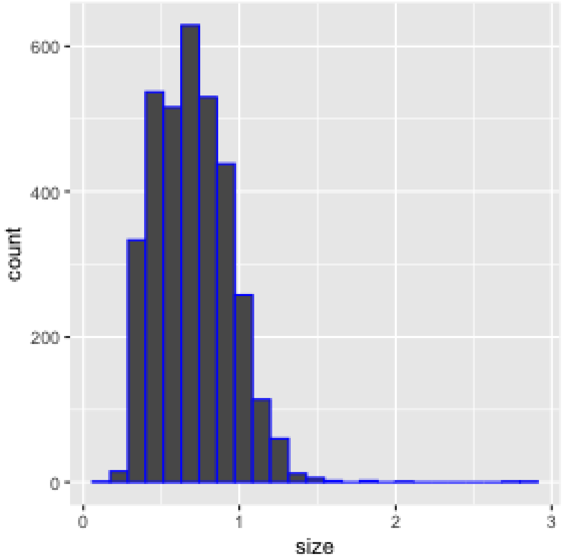


(A bar chart is appropriate when the x-axis variable is categorical and the y-axis variable is quantitative.)

ggplot: Histogram

What is the distribution of galaxy sizes?

```
ggplot(data=df,mapping=aes(x=size)) +
  geom_histogram(color="blue",bins=25)
```

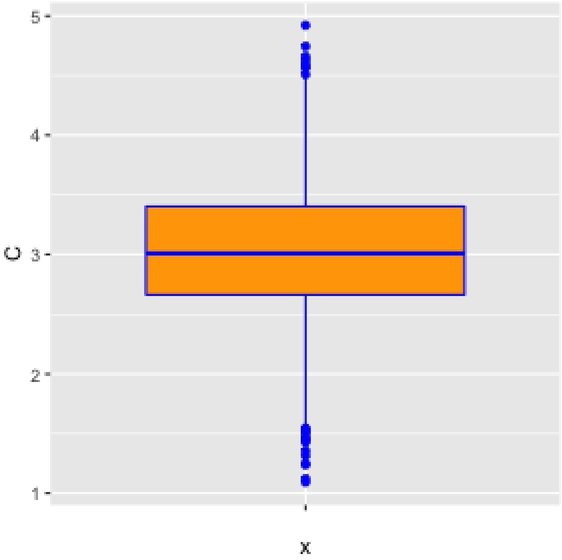


(A histogram is appropriate when the single variable in question is quantitative.)

ggplot: Boxplot

Boxplots are just a bit trickier. What is the distribution of galaxy concentrations?

```
ggplot(data=df,mapping=aes(x="",y=C)) +
  geom_boxplot(color="blue",fill="orange")
```

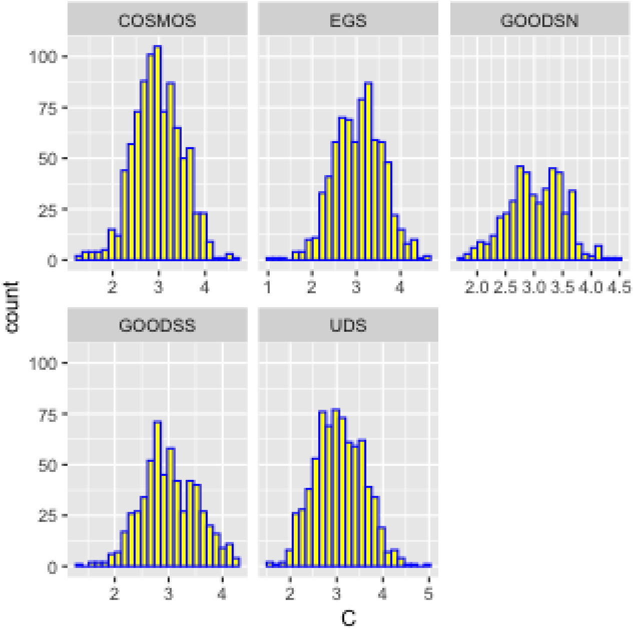


(A boxplot is appropriate when the single variable in question is quantitative.)

ggplot: Faceting

Faceting is the act of making multiple plots at once that appear side-by-side as "facets". Faceting is something you might want to do when, e.g., you have a factor variable. Here, we show histograms of the concentration variable `C` broken up by galaxy field.

```
ggplot(data=df,mapping=aes(x=C)) +
  geom_histogram(color="blue",fill="yellow",bins=25) +
  facet_wrap(~field,scales='free_x')
```



`free_x` means let the limits along the x-axes be different for each faceted plot.

ggplot: Gather

`gather()` is a function (from the `tidyr` package) that takes a data frame and realigns it. It is best illustrated via a simple example. Let's say we have the following data frame, which we'll call `df`:

```
> df
     x   y
1  0.5 0.7
2  1.2 1.9
```

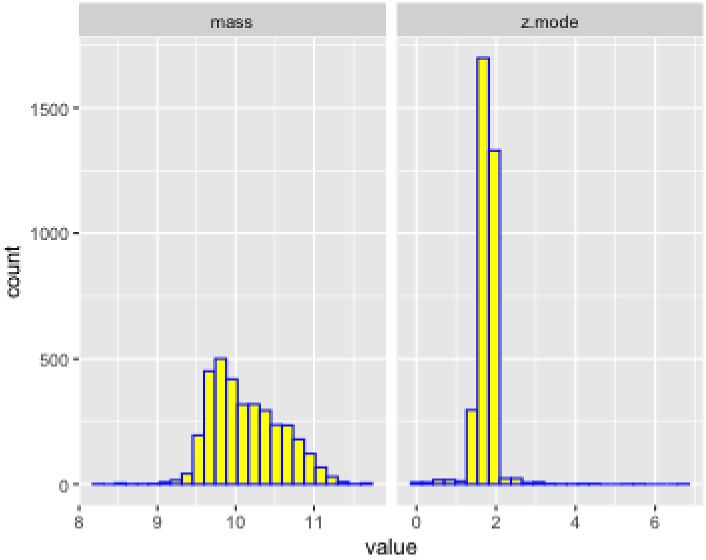
If we "gather" these data, we get the following:

```
> library(tidyr)
...
> gather(df)
      key  value
1       x    0.5
2       x    1.2
3       y    0.7
4       y    1.9
```

Combining `gather()` with faceting allows one to, e.g., visualize the data in multiple columns in a data frame at once, side by side.

ggplot: Gather (+ dplyr)

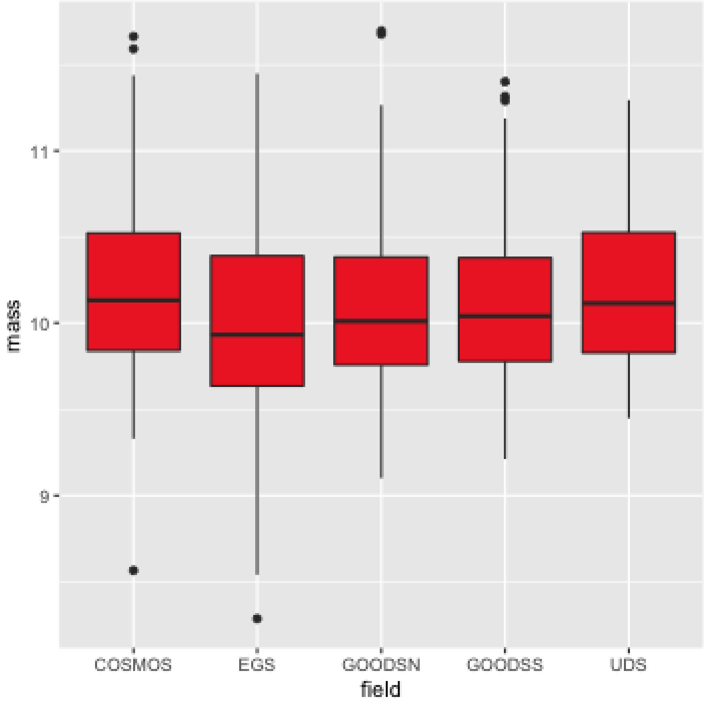
```
df.new <- df %>%
  dplyr::select(.,z.mode,mass) %>%
  gather(.)
#
# df.new has two columns: key and value
#
ggplot(data=df.new,mapping=aes(x=value)) +
  geom_histogram(color="blue",fill="yellow",bins=25) +
  facet_wrap(~key,scales='free_x')
```



(An unfortunate quirk that arises here is our typing `dplyr::select` above. This is due to a *namespace* issue. This notes file loads both the `dplyr` and `MASS` packages, and both have functions named `select`. The `dplyr::select` tells `R` to use the `select` function in `dplyr`.)

ggplot: Side-By-Side Boxplots

```
ggplot(data=df,mapping=aes(x=field,y=mass)) +
  geom_boxplot(fill="FireBrick2")
```



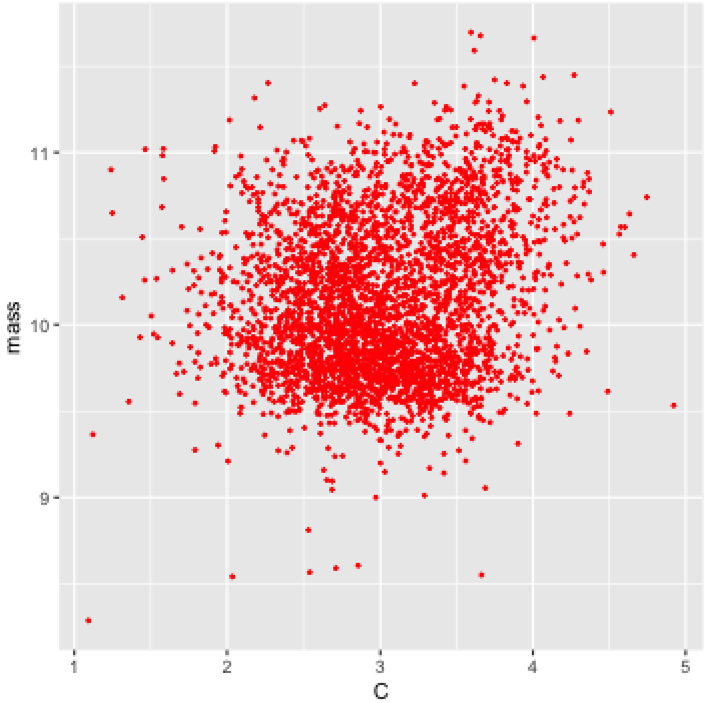
Note the color used above: `FireBrick2`. Where can you find a list of the color names that `R` recognizes? Right here.

Enjoy using `burlywood` and `darksalmon`, among other colors.

ggplot: Scatter Plot

A two-dimensional scatter plot allows us to get a sense of how the data in different columns are associated with one another. Here, let's plot `mass` vs. `C`:

```
ggplot(data=df,mapping=aes(x=C,y=mass)) +
  geom_point(color="red",size=0.5)
```

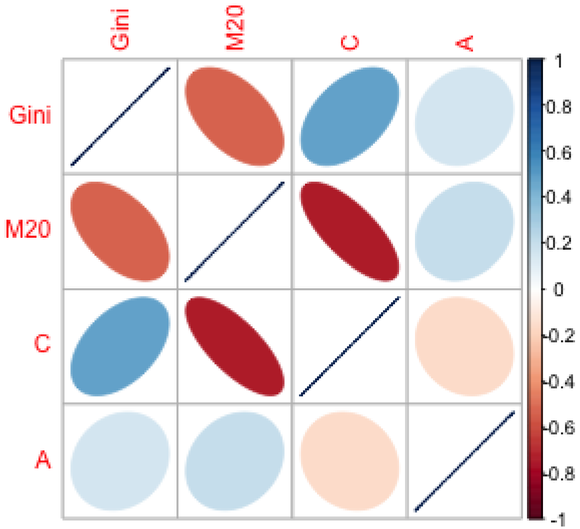


Beyond ggplot: corrplot

Analysts often use scatter plots to visually assess the level of *correlation*, or *linear dependence*, between the data in two columns of a data frame. (Recall that if two variables are "uncorrelated," it does not mean that they are "independent"...the latter means there is no dependence, linear or otherwise, between two variables.)

Another way to visualize correlations is `corrplot`:

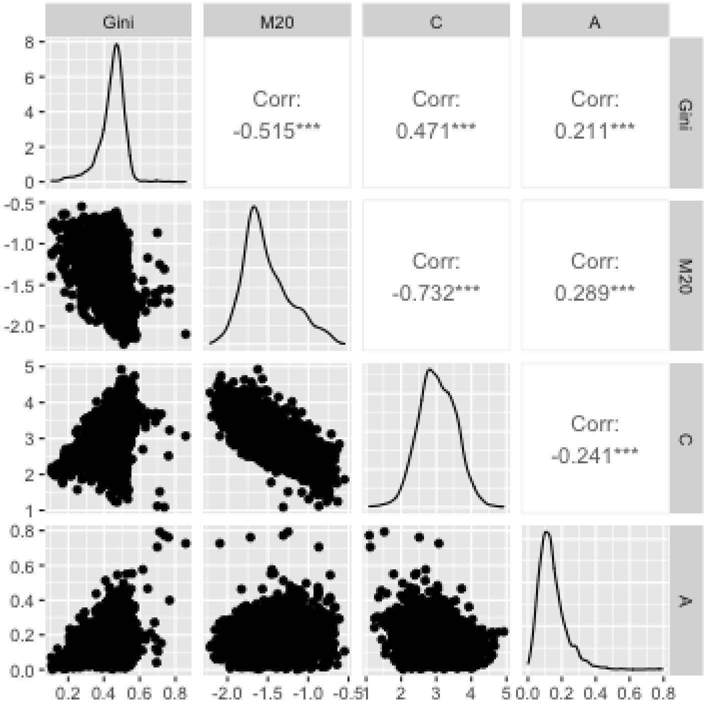
```
suppressMessages(library(corrplot))
# Remember: correlation values range from -1 (negatively correlated) to +1 (positively correlated)
df %>%
  dplyr::select(.,Gini,M20,C,A) %>%
  cor(.) %>%
  corrplot(.,method="ellipse")
```



Beyond ggplot: ggpairs

Base R provides the pairs() function, which creates a matrix of scatterplots. A version of the pairs() function, on steroids, is ggpairs:

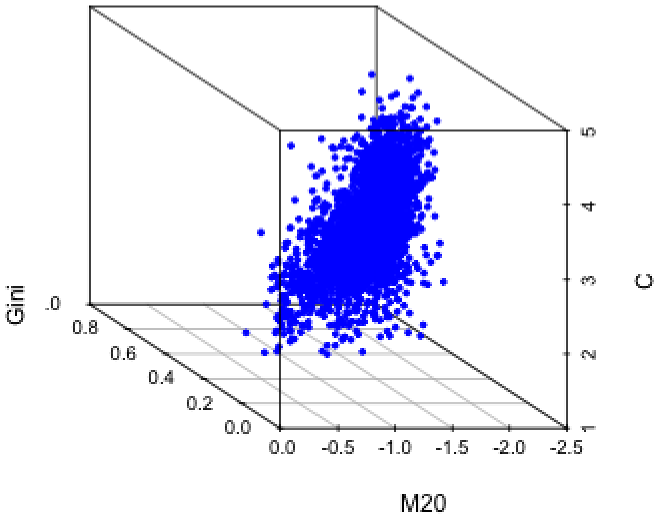
```
suppressMessages(library(GGally))
df %>%
  dplyr::select(.,Gini,M20,C,A) %>%
  filter(.,A>0) %>%
  ggpairs(.,progress=FALSE,lower=list(combo=wrap("facethist", binwidth=0.8)))
```



Beyond ggplot: scatterplot3d

One way to visualize data in three dimensions is via the `scatterplot3d` function:

```
library(scatterplot3d)
scatterplot3d(x=df$Gini,y=df$M20,z=df$C,pch=19,
              color="blue",angle=-30,cex.symbols=0.5,
              xlab="Gini",ylab="M20",zlab="C")
```



Beyond ggplot: parcoord

The `parcoord()` function is a mechanism through which we can attempt to visualize more than three variables at once. Each line represents a single object, i.e., a single row of a data frame.

```
suppressMessages(library(MASS))
z.color <- round(64*(df$Gini-min(df$Gini))/(max(df$Gini)-min(df$Gini)))
palette(rainbow(64))
df %>%
  dplyr::select(.,Gini,M20,C,A) %>%
  filter(.,A>0) %>%
  parcoord(.,col=z.color[df$A>0],lwd=0.4)
```