Visualization with ggplot2

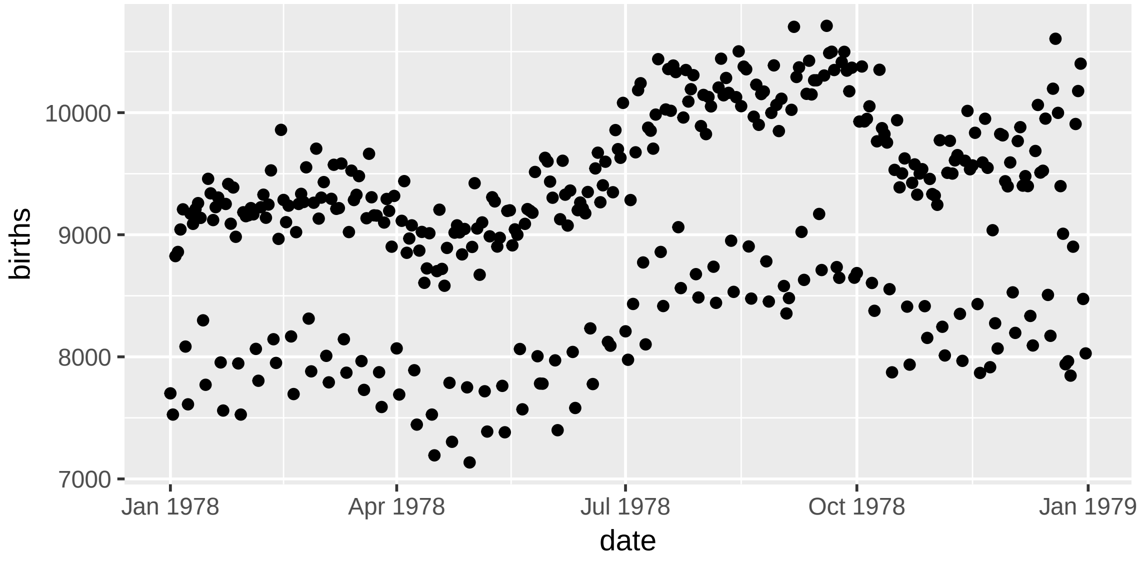
The grammar of graphics

- geom: the geometric "shape" used to display data (glyph)
- aesthetic: an attribute controlling how geom is displayed (Tufte called this a visual cue)
- scale: conversion of raw data to visual display (particular assignment of colors, shapes, sizes, etc.)
- guide: helps user convert visual data back into raw data (legends, axes)
- stat: a transformation applied to data before geom gets it (an example is that a histogram works on binned data)

Set-up

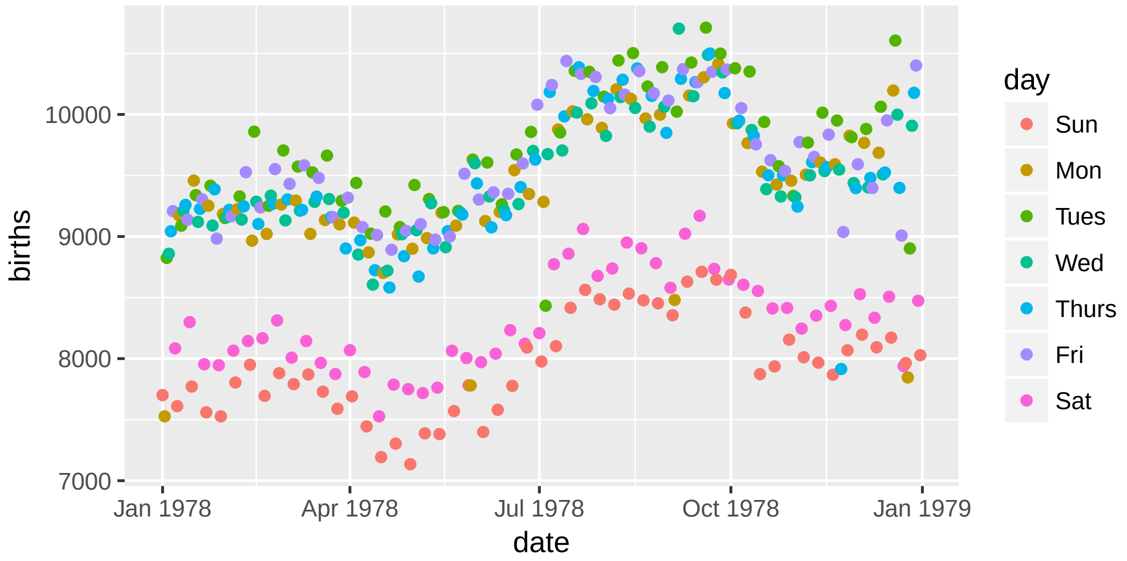
```
require(mosaic)
require(lubridate) # package for working with dates
data(Births78)
head(Births78)

##         date births dayofyear
## 1 1978-01-01   7701         1
## 2 1978-01-02   7527         2
## 3 1978-01-03   8825         3
## 4 1978-01-04   8859         4
## 5 1978-01-05   9043         5
## 6 1978-01-06   9208         6
```

How do we make this plot?



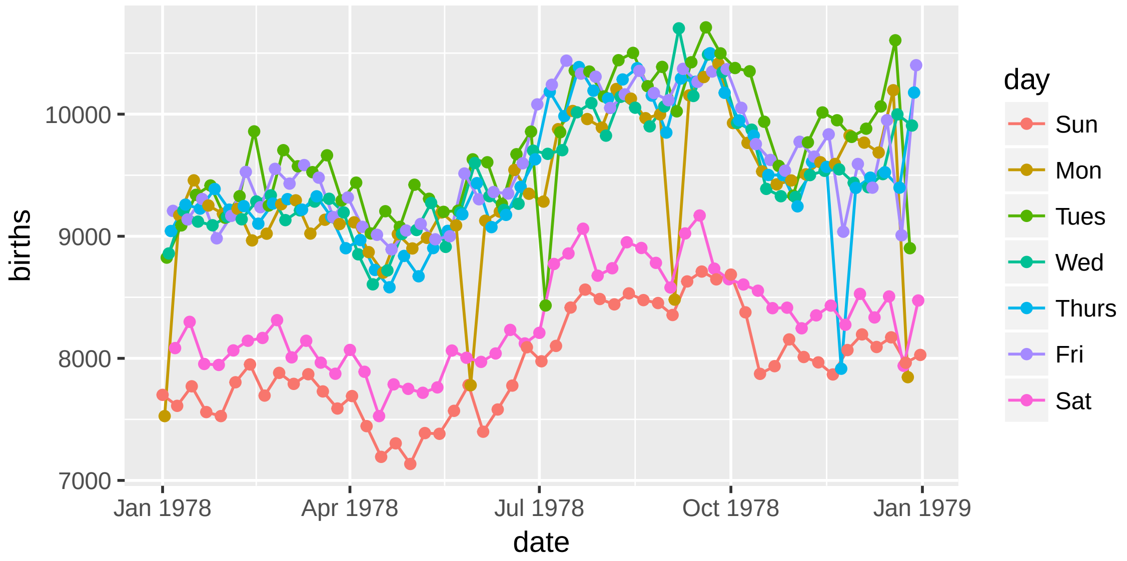
```
require(ggplot2)
ggplot(data=Births78) +
  geom_point(aes(x=date, y=births))
```

How do we make this plot?



```
Births78=mutate(Births78, day = wday(date, label=TRUE))
ggplot(data=Births78) +
  geom_point(aes(x=date, y=births, color=day))
```

How do we make this plot?



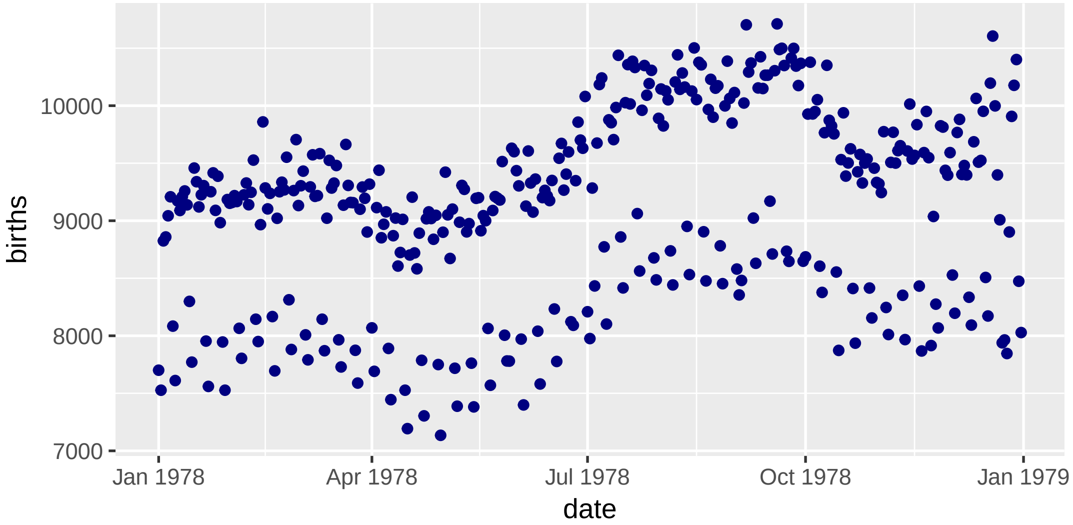
```
ggplot(data=Births78)+
  geom_line(aes(x=date, y=births, color=day))+
  geom_point(aes(x=date, y=births, color=day))
```

What does this code do?

```
ggplot(Births78)+
  geom_point(aes(x=date, y=births, color="navy"))
```

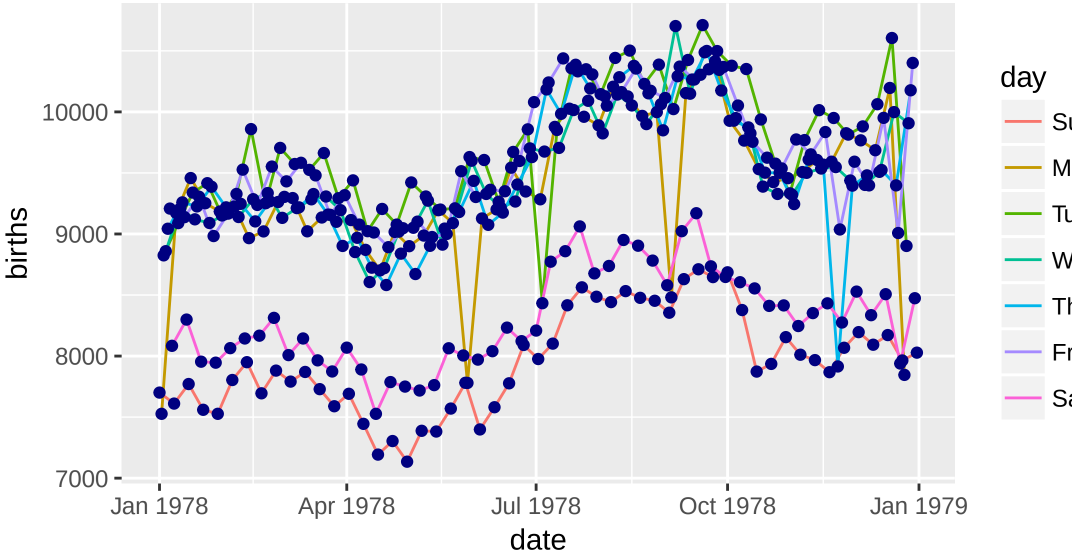
- This is mapping the color aesthetic to a new variable with only one value (navy).
- So all the dots get set to the same color, but its not navy.
- Mapping vs. Setting

```
ggplot(data=Births78)+
  geom_point(aes(x=date, y=births), color="navy")
```



- Note that (color = ”navy”) is now outside of the aesthetics list. Thats how ggplot2 distinguishes between mapping and setting.

How do we make this plot?



```
ggplot(data=Births78) +
  geom_line(aes(x=date, y=births, color=day)) + # map color here
  geom_point(aes(x=date, y=births),color="navy") # set color here
```

Some Notes

- ggplot() establishes the default data and aesthetics for the geoms, but each geom may change these defaults.
- good practice: put into ggplot the things that affect all (or most) of the layers; rest in `geom_blah`.

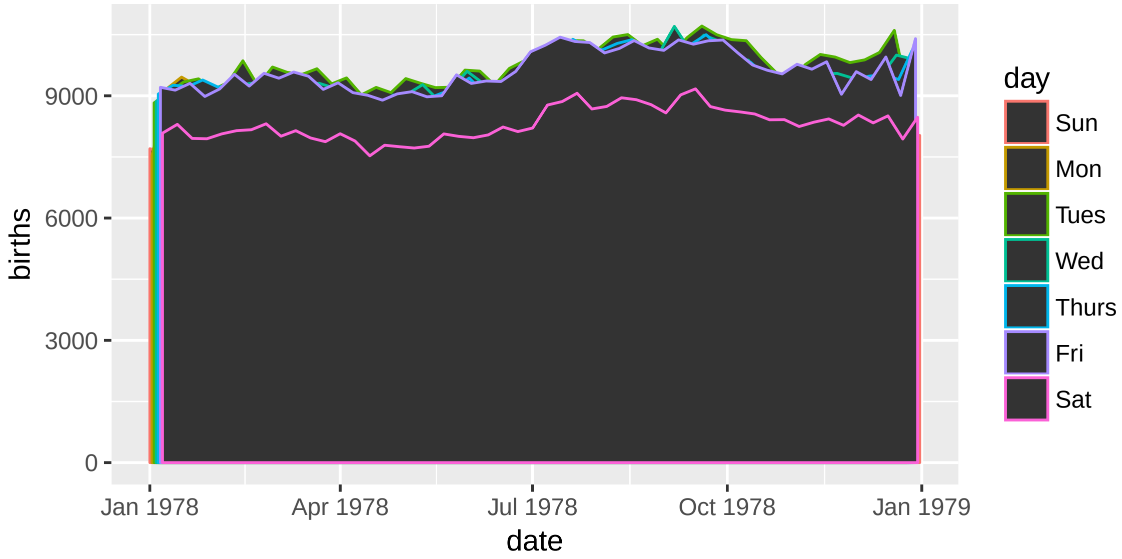
What other geoms are there?

```
apropos("^geom_")

##  [1] "geom_abline"     "geom_area"       "geom_ash"
##  [4] "geom_bar"        "geom_bin2d"      "geom_blank"
##  [7] "geom_boxplot"    "geom_col"        "geom_contour"
## [10] "geom_count"      "geom_crossbar"   "geom_curve"
## [13] "geom_density"    "geom_density_2d" "geom_density2d"
## [16] "geom_dotplot"    "geom_errorbar"   "geom_errorbarh"
## [19] "geom_freqpoly"   "geom_hex"        "geom_histogram"
## [22] "geom_hline"      "geom_jitter"     "geom_label"
## [25] "geom_line"       "geom_linerange"  "geom_map"
## [28] "geom_path"       "geom_point"      "geom_pointrange"
## [31] "geom_polygon"    "geom_qq"         "geom_quantile"
## [34] "geom_raster"     "geom_rect"       "geom_ribbon"
## [37] "geom_rug"        "geom_segment"    "geom_smooth"
## [40] "geom_spline"     "geom_spoke"      "geom_step"
## [43] "geom_text"       "geom_tile"       "geom_violin"
## [46] "geom_vline"
```

- Help pages will tell you their aesthetics, default stats, etc.

Let's try `geom_area`

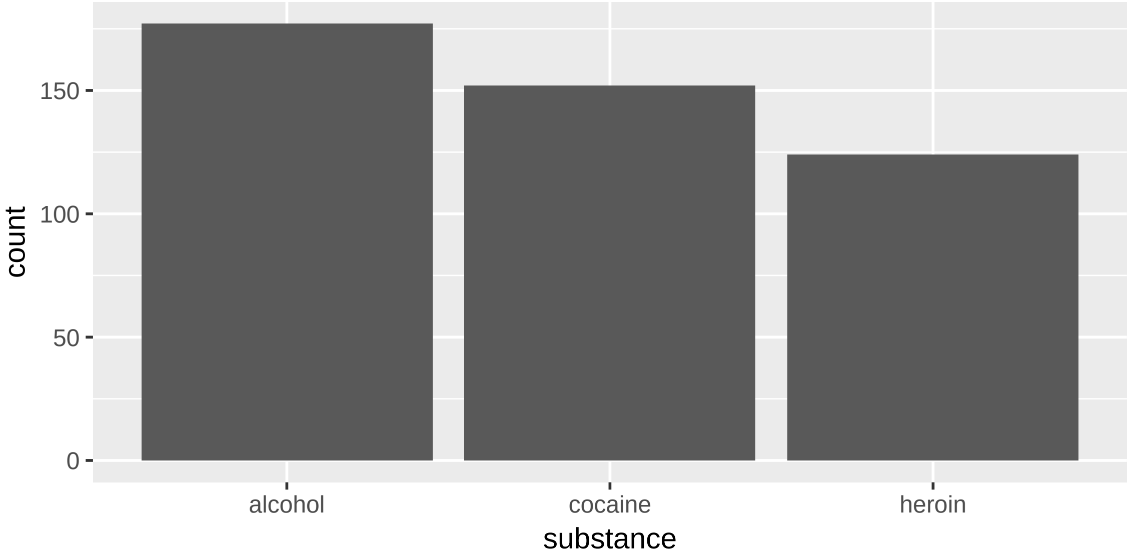
```
ggplot(data=Births78) +
  geom_area(aes(x=date, y=births, color=day))
```



- This is not a good plot
  - overplotting is hiding much of the data
  - extending y-axis to 0 may or may not be desirable.

HELPrct data set Why are people in the study?

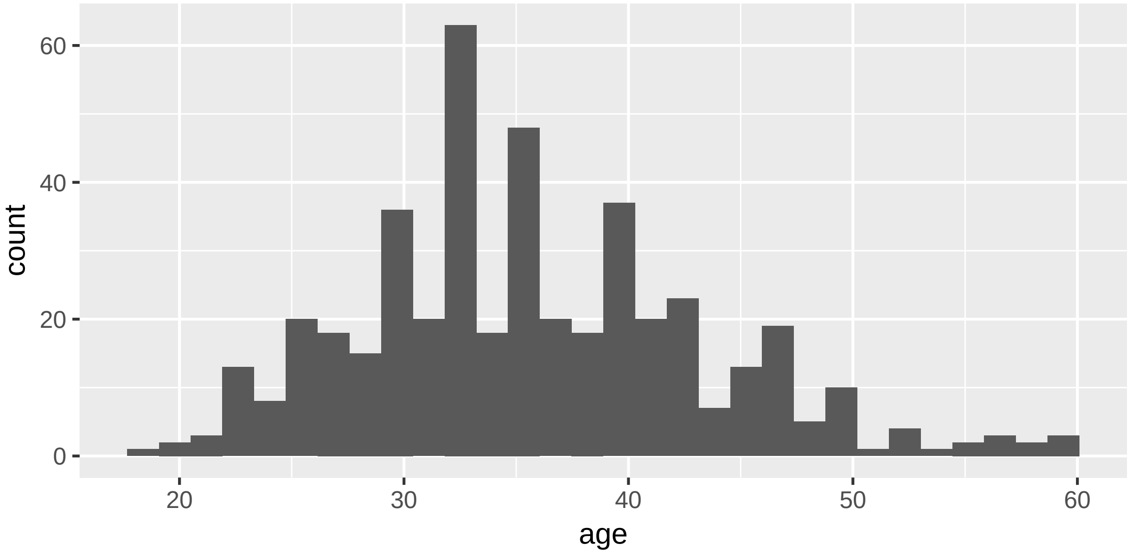
```
ggplot(data=HELPrct) +
  geom_bar(aes(x=substance))
```



- `stat_bin()` is being applied to the data before the `geom_bar()` gets to do its thing. Binning creates the y values automatically.

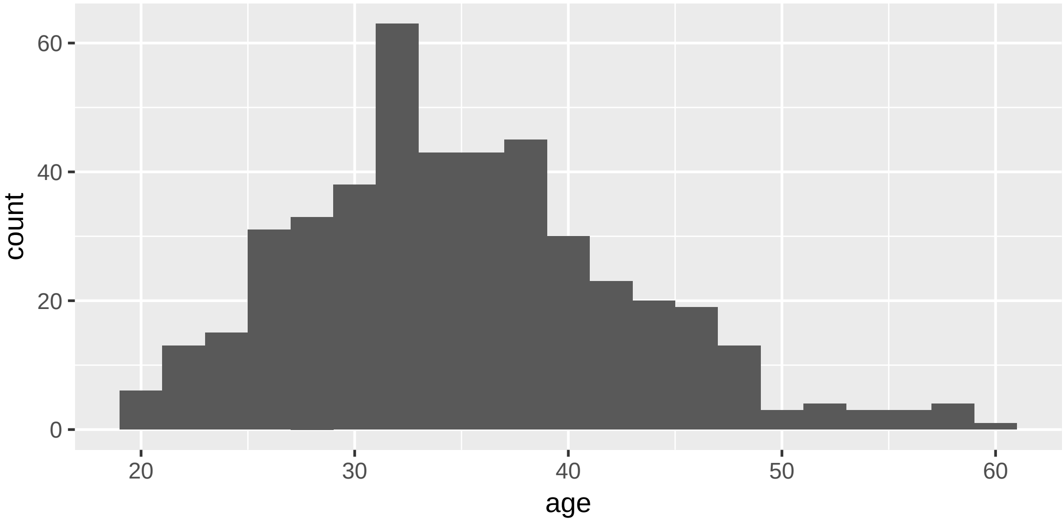
How old are people in the study?

```
ggplot(data=HELPrct) +
  geom_histogram(aes(x=age))

## `stat_bin()` using `bins = 30`.  Pick better value with `binwidth`.
```



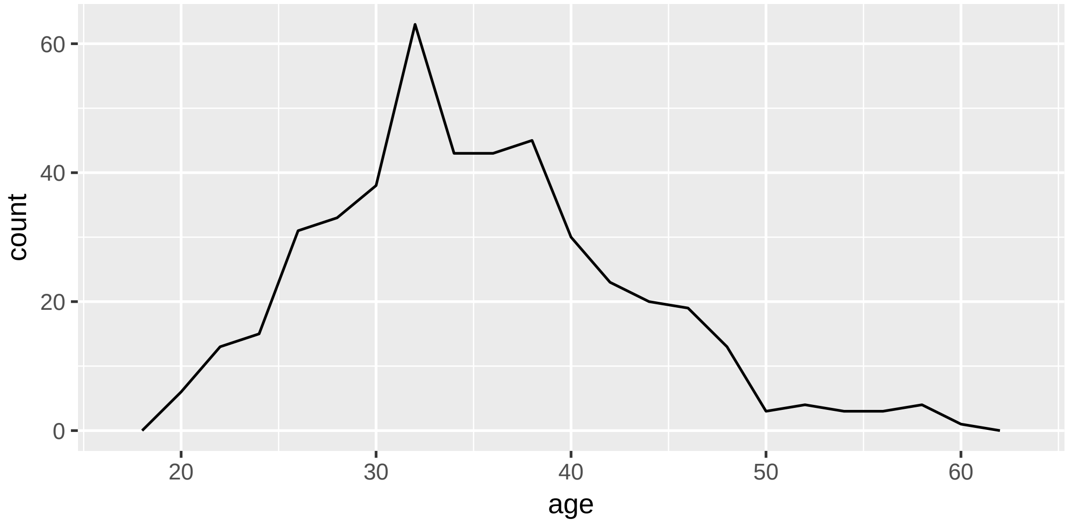
- Notice the message, we can adjust the binwidth manually.

```r
ggplot(data=HELPrct) +
  geom_histogram(aes(x=age), binwidth=2)
```

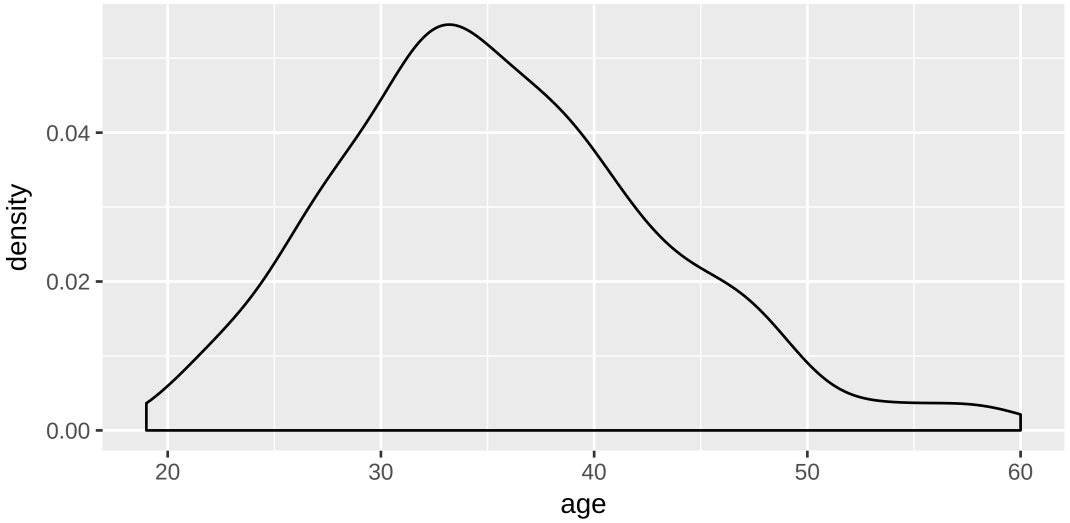


- Let's take a look at other geoms.

```r
ggplot(data=HELPrct) +
  geom_freqpoly(aes(x=age),binwidth=2)
```

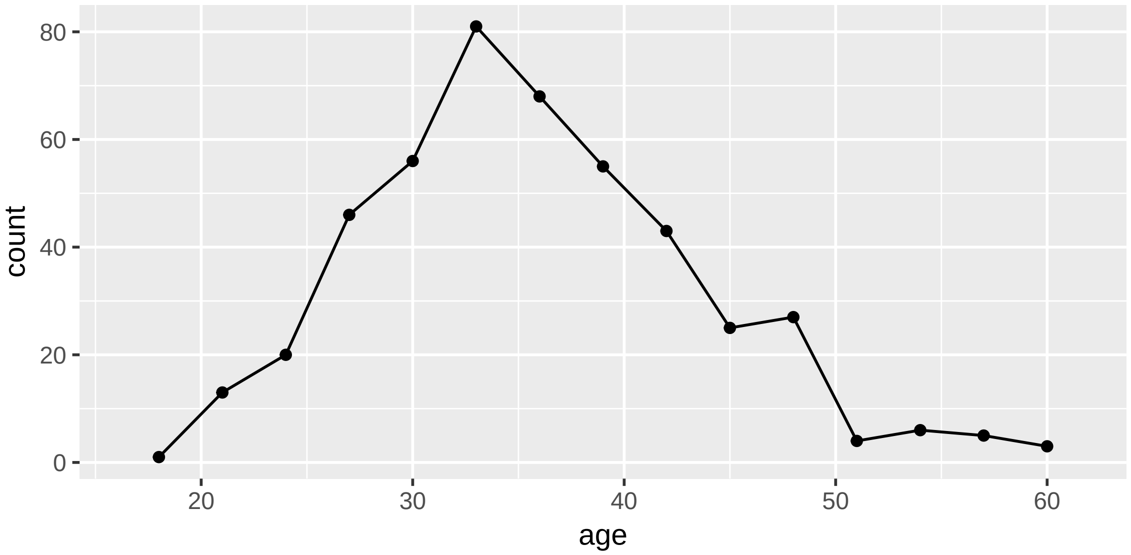


```
ggplot(data=HELPrct) +
  geom_density(aes(x=age))
```

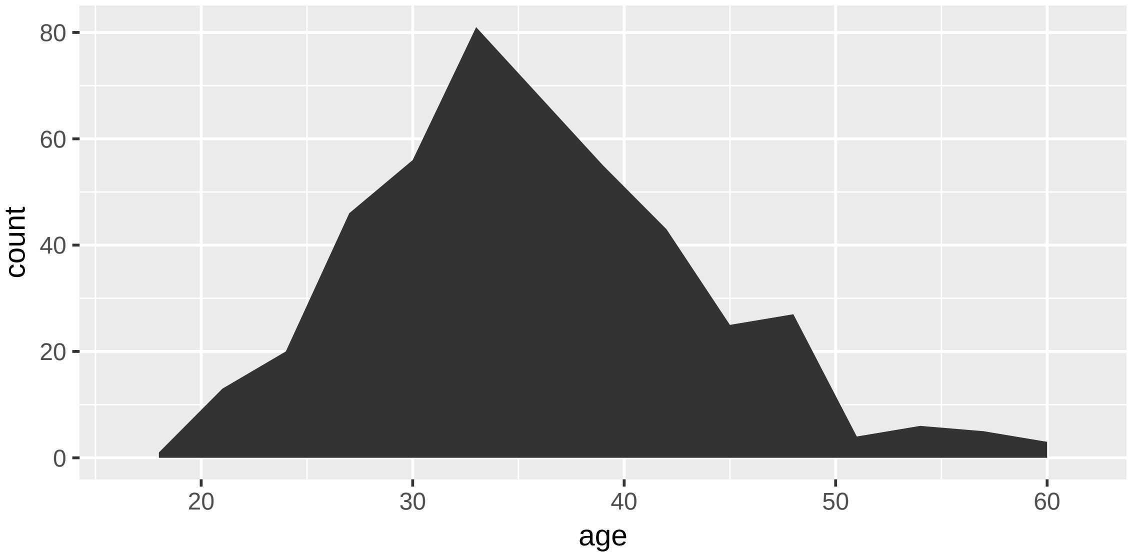


- Every stat comes with a default geom, most geoms come with a default stat
  - we can specify stats instead of geom, if we prefer
  - we can mix and match geoms and stats however we like

```
ggplot(data=HELPrct) +
  geom_point(aes(x=age), stat="bin", binwidth=3) +
  geom_line(aes(x=age), stat="bin", binwidth=3)
```


80
60
40
20
0
count
20
30
40
50
60
age

```
ggplot(data=HELPrct) +
  geom_area(aes(x=age), stat="bin", binwidth=3)
```



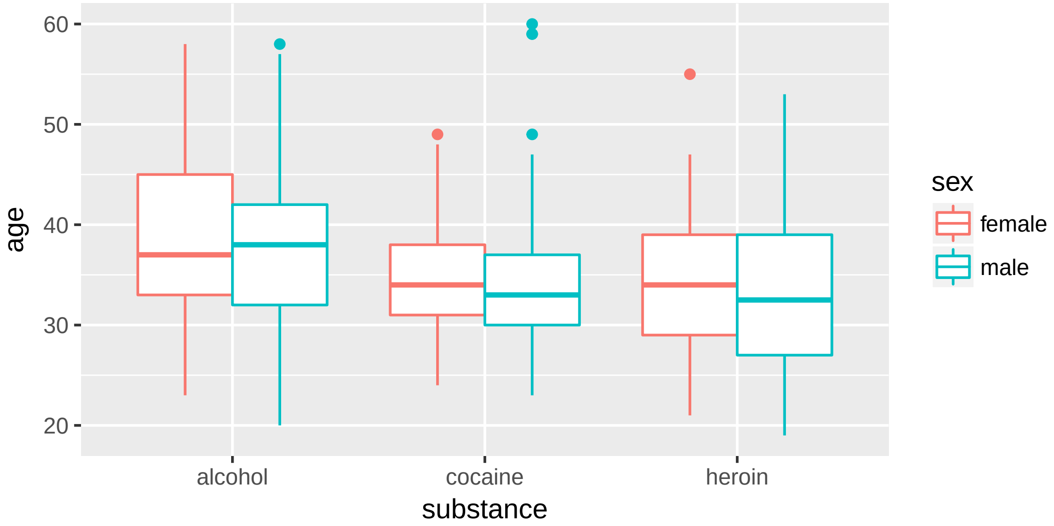
You Try

- Create a plot (or two) that shows the distribution of the average daily alcohol consumption in the past 30 days (i1).
- Covariates: Adding in more variables
  - How does alcohol consumption (or age, your choice) differ by sex and substance (alcohol, cocaine, heroin)?
    - Decisions:
      - How will we display the variables: i1 (or age), sex, substance?
      - What comparisons are we most interested in?
    - Give it a try: You may want to do some things I havent shown you yet. (Feel free to ask.)

Boxplots

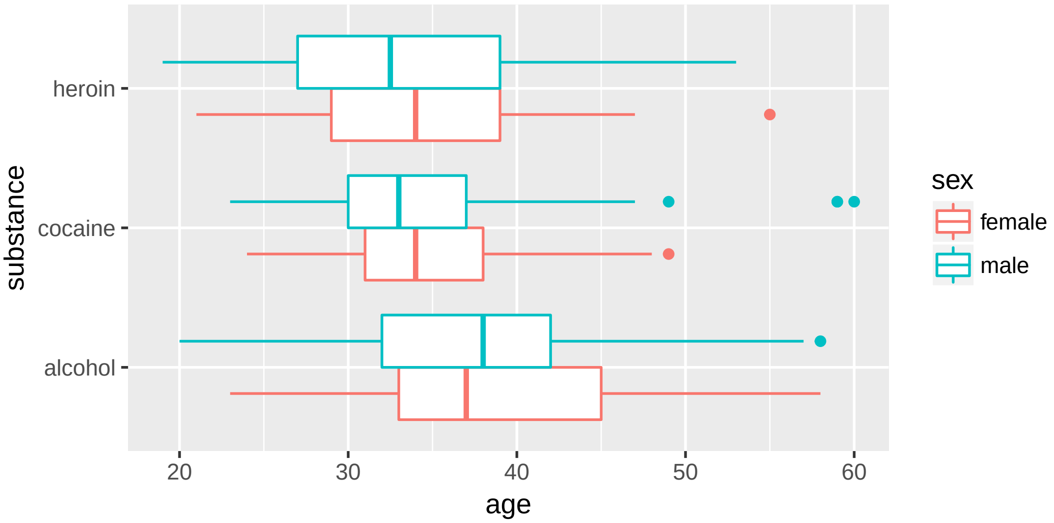
- Boxplots use `stat_quantile()` which computes a five-number summary (roughly the five quartiles of the data) and uses them to define a box and whiskers. The quantitative variable must be y, and there must be an additional categorical x variable.

```
ggplot(data=HELPrct) +
  geom_boxplot(aes(x=substance, y=age, color=sex))
```



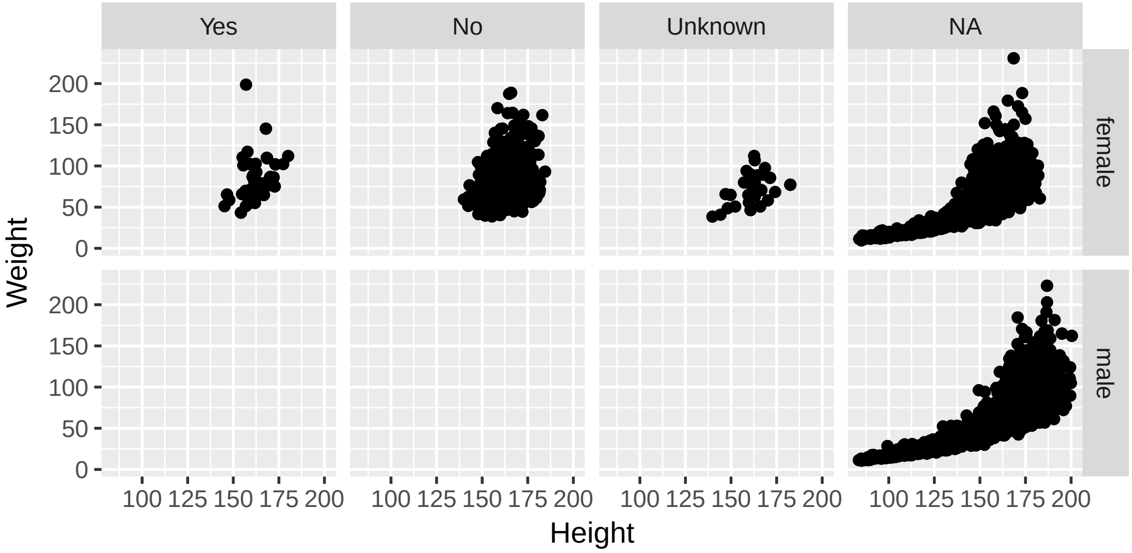
- Horizontal boxplots are obtained by flipping the coordinate system:

```
ggplot(data=HELPrct) +
  geom_boxplot(aes(x=substance, y=age, color=sex)) +
  coord_flip()
```



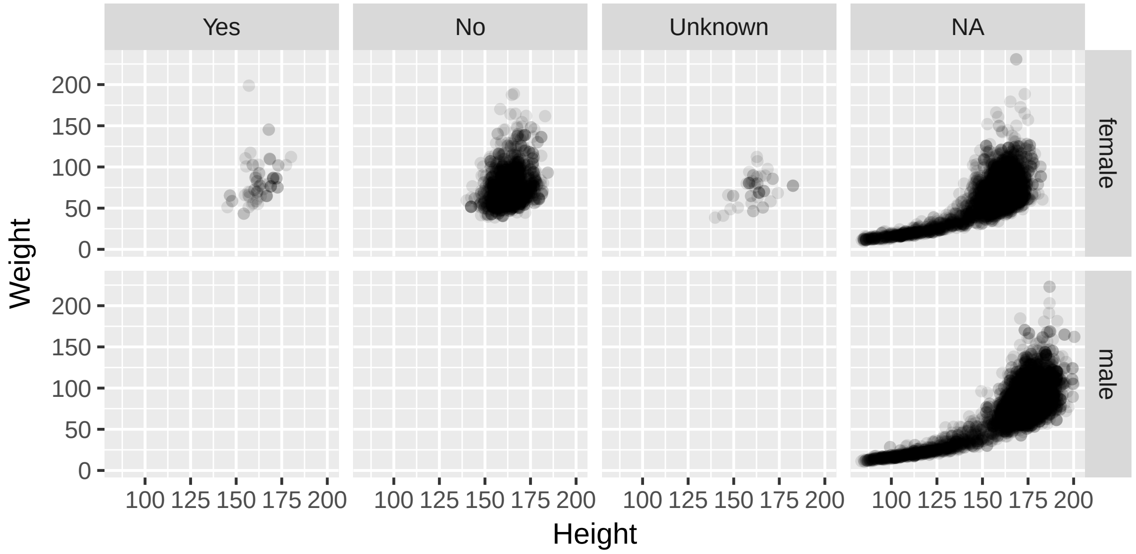
Issues with Bigger Data

```
require(NHANES)
dim(NHANES)

## [1] 10000    76

ggplot(data=NHANES) +
  geom_point(aes(x=Height, y=Weight)) +
  facet_grid( Gender ~ PregnantNow )
```



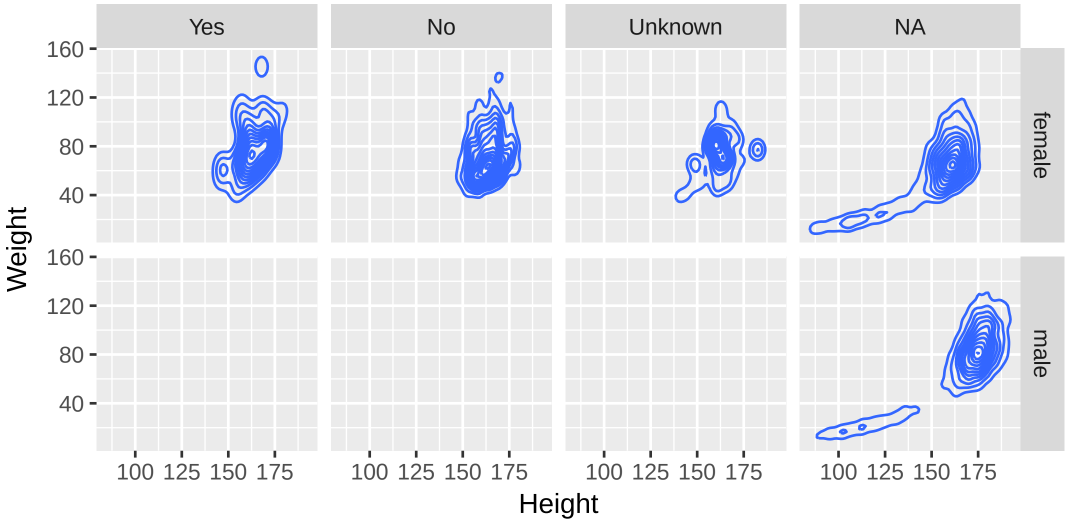
- Although we can see a generally positive association, it is very difficult to see where most of the data lies.
- One way to deal with overplotting is to set the opacity low.

```
ggplot(data=NHANES) +
  geom_point(aes(x=Height, y=Weight), alpha=0.1) +
  facet_grid( Gender ~ PregnantNow )
```



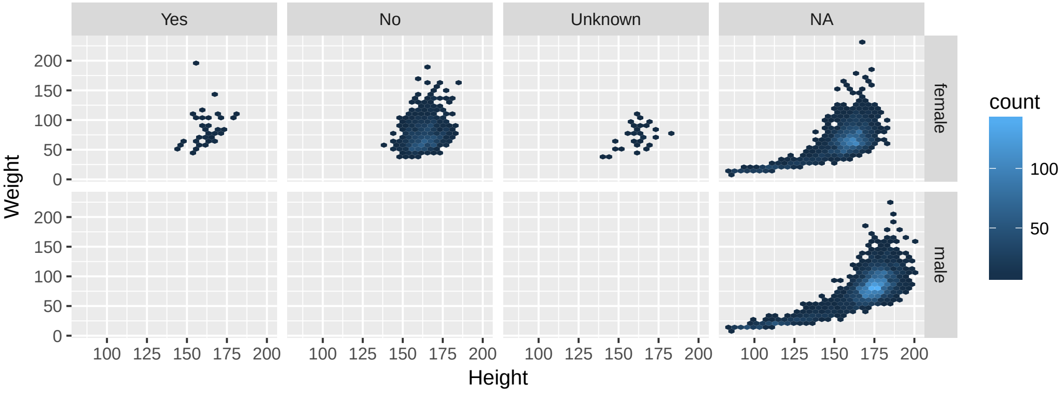
- Or we could try an entirely different geom

```
ggplot(data=NHANES) +
  geom_density2d(aes(x=Height, y=Weight)) +
  facet_grid( Gender ~ PregnantNow )
```



- Or maybe you prefer

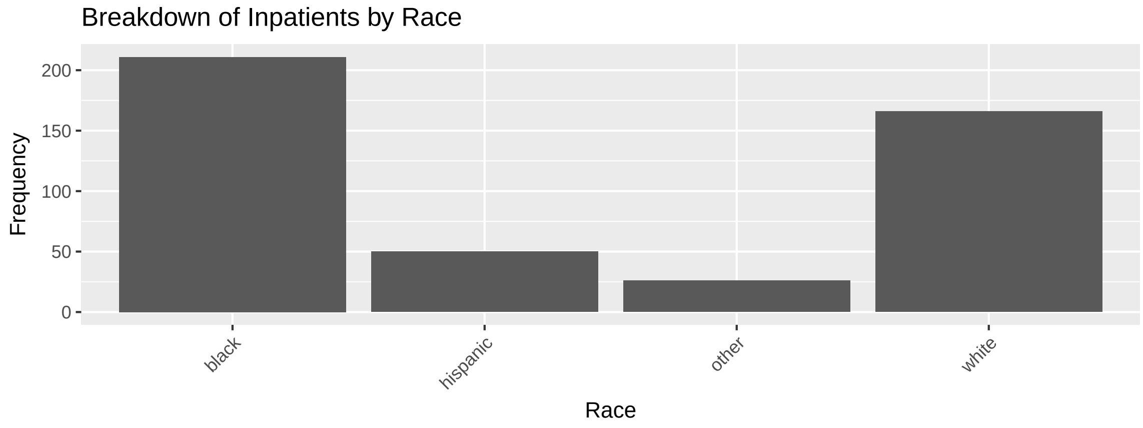
```
#install.packages("hexbin")
require(hexbin)

## Loading required package:  hexbin

ggplot(data=NHANES) +
  geom_hex(aes(x=Height, y=Weight)) +
  facet_grid( Gender ~ PregnantNow )
```


Yes
No
Unknown
NA
female
male
Weight
Height
count
100
50
0
50
100
150
200
100
125
150
175
200

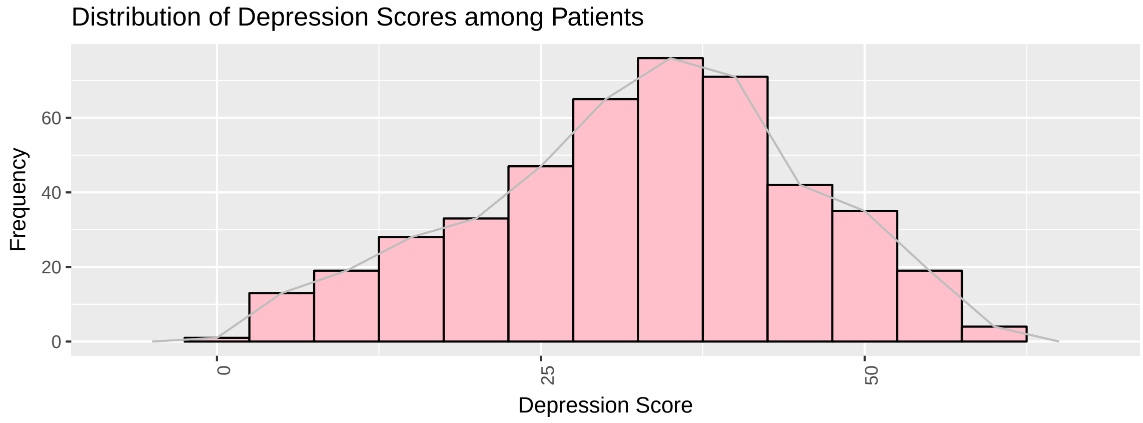
Titles and Labels

We just need to add more layers to a basic plot to achieve titles and axis labels. The theme() layer is not necessary. The customization we are making here is that the tick labels on the x axis are being rotated 45 degrees. Without this option, the tick labels on the x axis would just be parallel to the x axis.

```
ggplot(data=HELPrct)+
    geom_bar(aes(x=racegrp))+
    ggtitle("Breakdown of Inpatients by Race")+
    xlab("Race")+
    ylab("Frequency")+
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
ggplot(data=HELPrct)+
    geom_histogram(aes(x=cesd), fill="pink", col="black", binwidth=5)+
    geom_freqpoly(aes(x=cesd), col="grey",binwidth=5)+
    ggtitle("Distribution of Depression Scores among Patients")+
    xlab("Depression Score")+
    ylab("Frequency")+
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```


Distribution of Depression Scores among Patients
Frequency
Depression Score