Class 6 - Plotting

GO GO GGPLOT

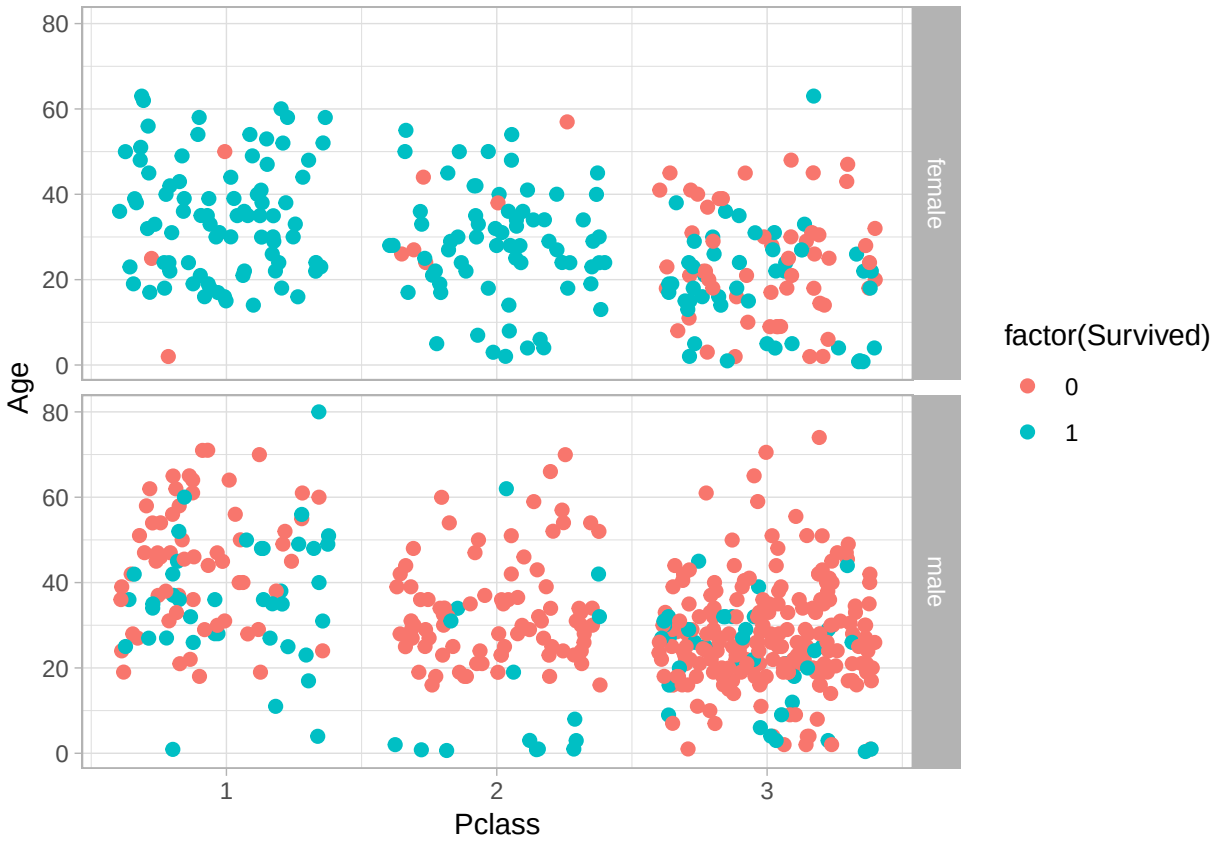
`ggplot` plots are creating by adding lines of code that each modify some part of the graph - colors, layout, axis ranges, text, etc. Remeber that in a plot, each thing you want to plot needs to be in its own column!

Here's a useful sheet for many different types of ggplots (they all follow the same format): http://www.sthda.com/english/wiki/be-awesome-in-ggplot2-a-practical-guide-to-be-highly-effective-r-software-and-data-visualization http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

Plotting for exploration

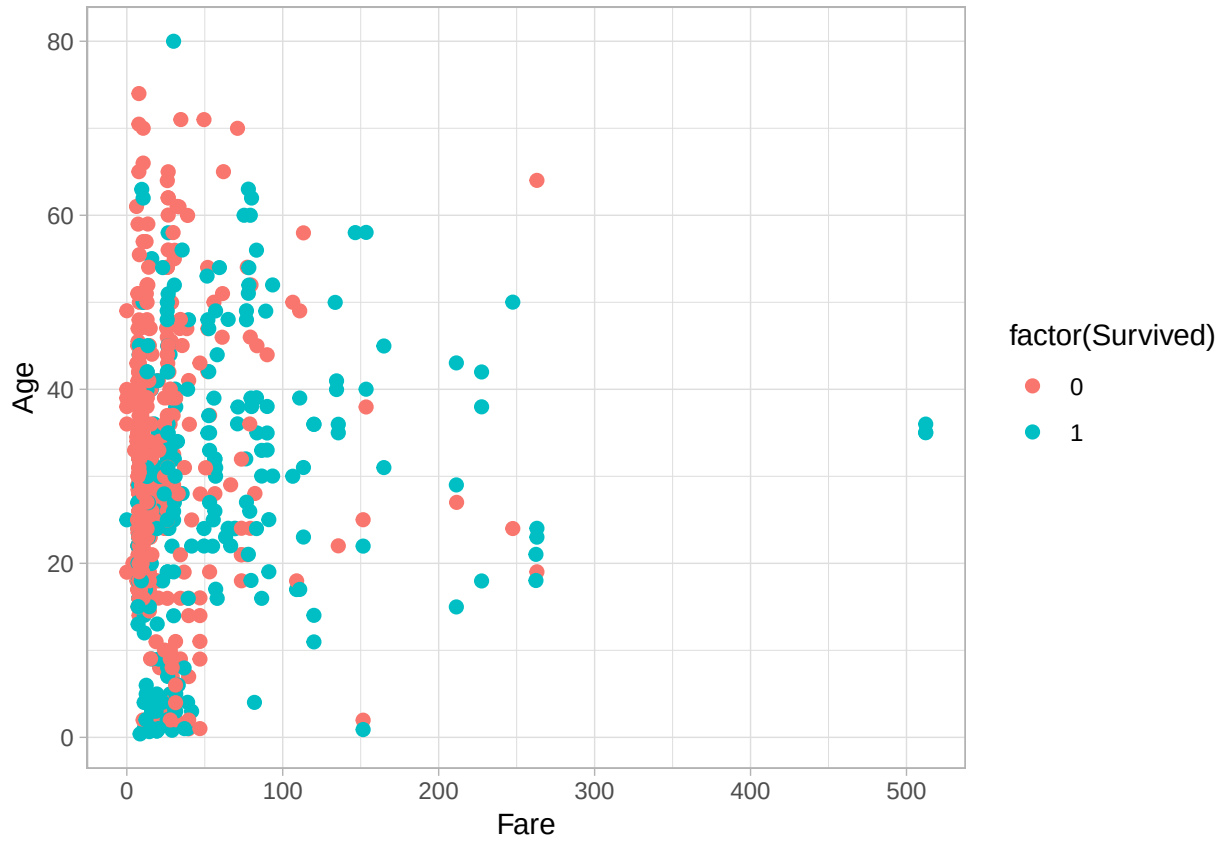
```
suppressMessages(library(ggplot2))
################################################################################
## let's use the titanic data to start
library(titanic)
titanic_data <- titanic_train


## what's the relationship between Pclass and Age, and how is this different by sex?
ggplot(data = titanic_data, aes(x = Pclass, y = Age, colour = factor(Survived))) +
    geom_jitter(size = 2) +
    facet_grid(Sex ~ .) +
    theme_light()
```

```
## Warning: Removed 177 rows containing missing values (geom_point).
```




```
## what's the relationship between fare and age?
ggplot(data = titanic_data, aes(x= Fare, y = Age, colour = factor(Survived))) +
    geom_jitter(size = 2) +
    theme_light()
```

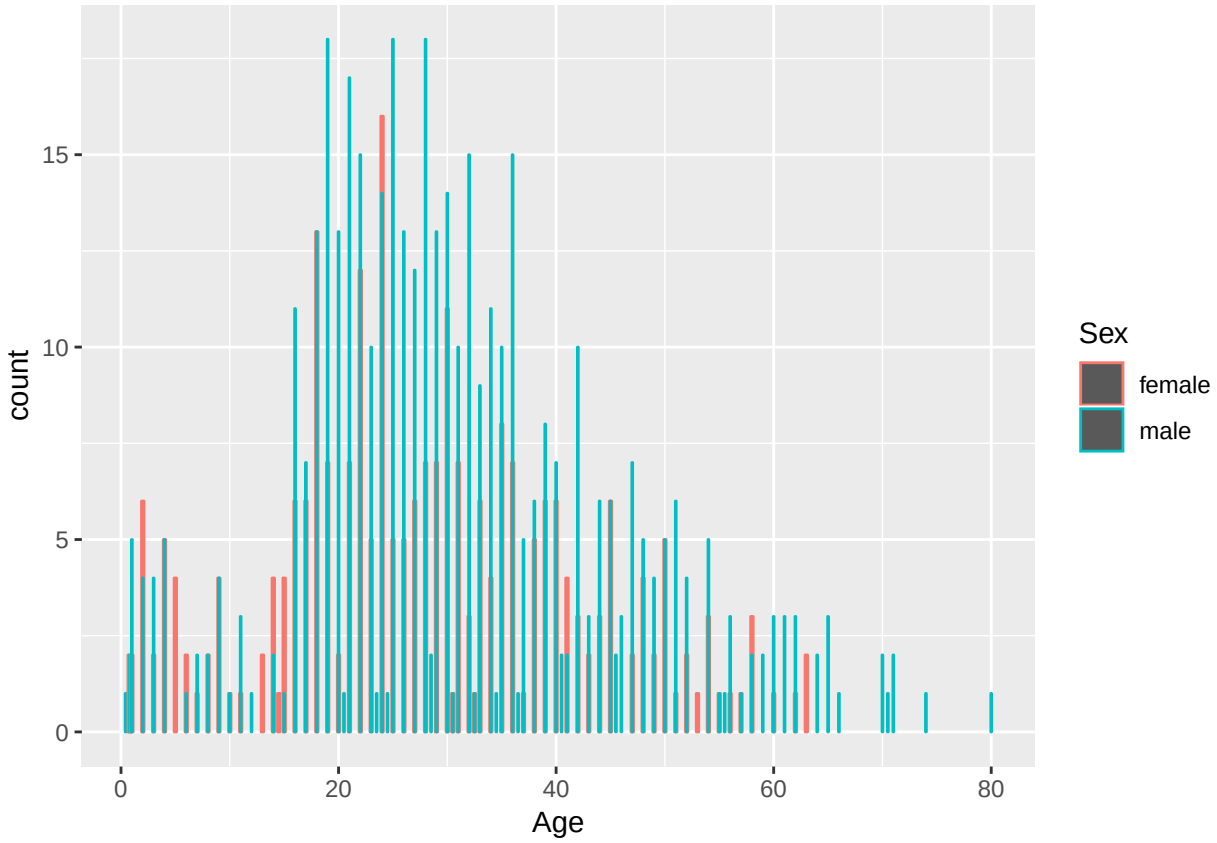
```
## Warning: Removed 177 rows containing missing values (geom_point).
```




```
## what is the distribution of ages, broken out by sex? here are some fun examples!
ggplot(titanic_data, aes(x = Age,  color = Sex)) +
  geom_bar()
```

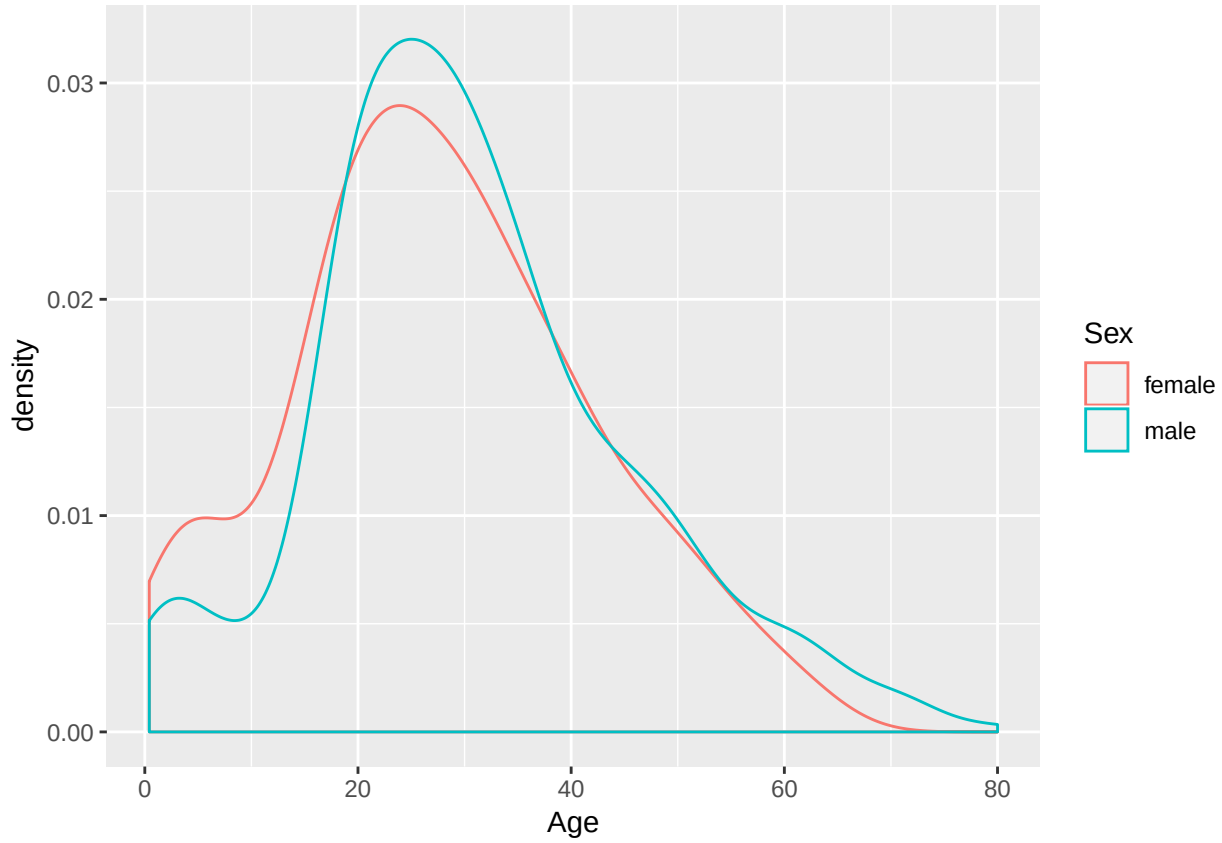
```
## Warning: Removed 177 rows containing non-finite values (stat_count).
```

```
## Warning: position_stack requires non-overlapping x intervals
```



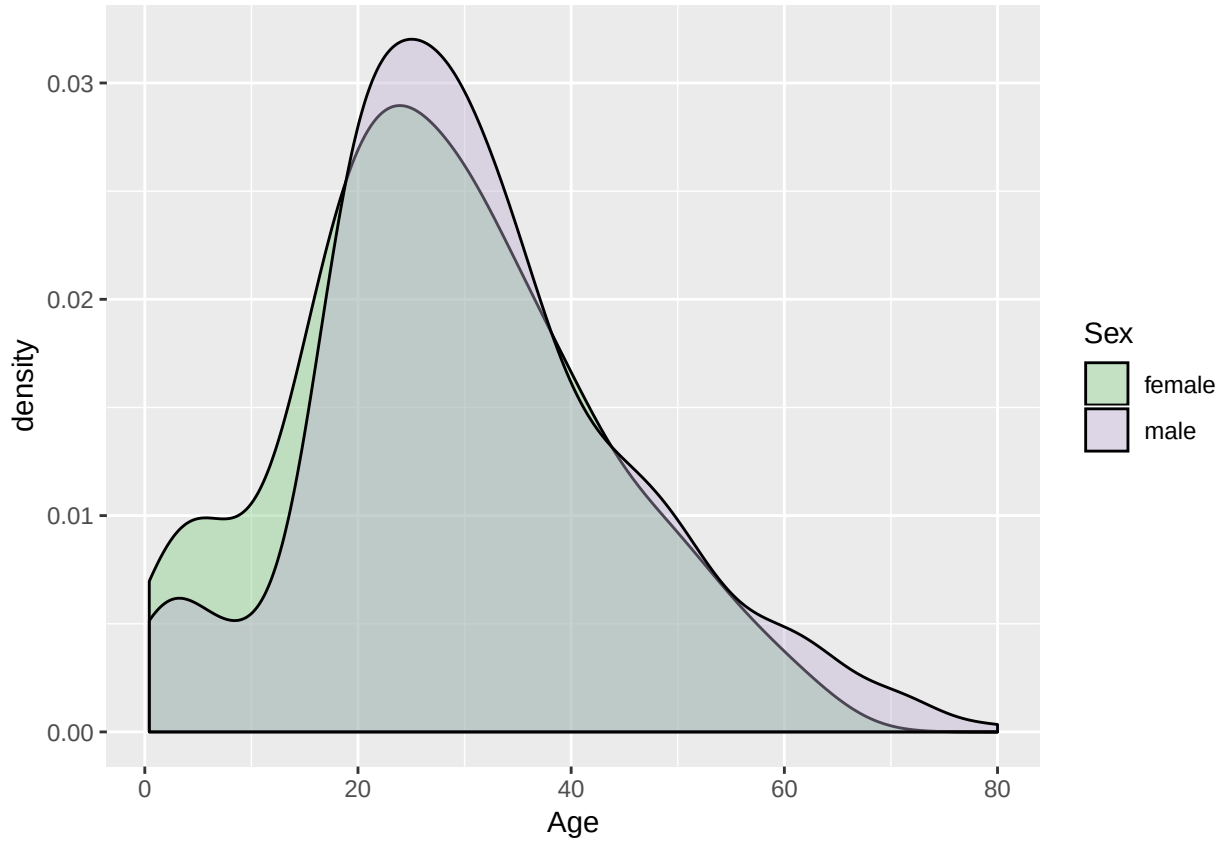

```
ggplot(titanic_data, aes(x = Age,  color = Sex)) +
  geom_density()
```

```
## Warning: Removed 177 rows containing non-finite values (stat_density).
```



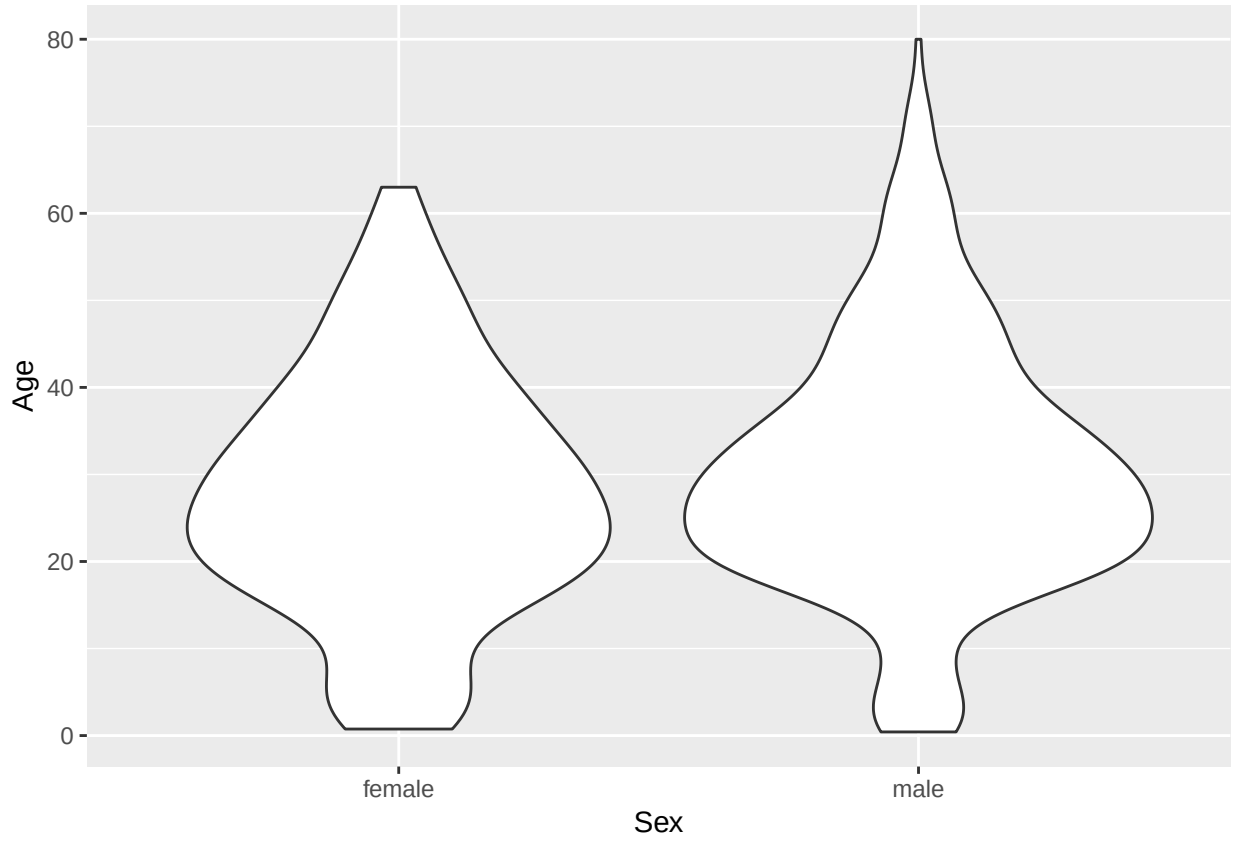

```
ggplot(titanic_data, aes(x = Age,  fill = Sex)) +
  geom_density(alpha = .4) +
  scale_fill_brewer(palette="Accent")
```

```
## Warning: Removed 177 rows containing non-finite values (stat_density).
```




```
## check out the color palettes here - https://www.r-graph-gallery.com/38-rcolorbrewers-palettes/

ggplot(titanic_data, aes(x = Sex,  y = Age)) +
  geom_violin()
```

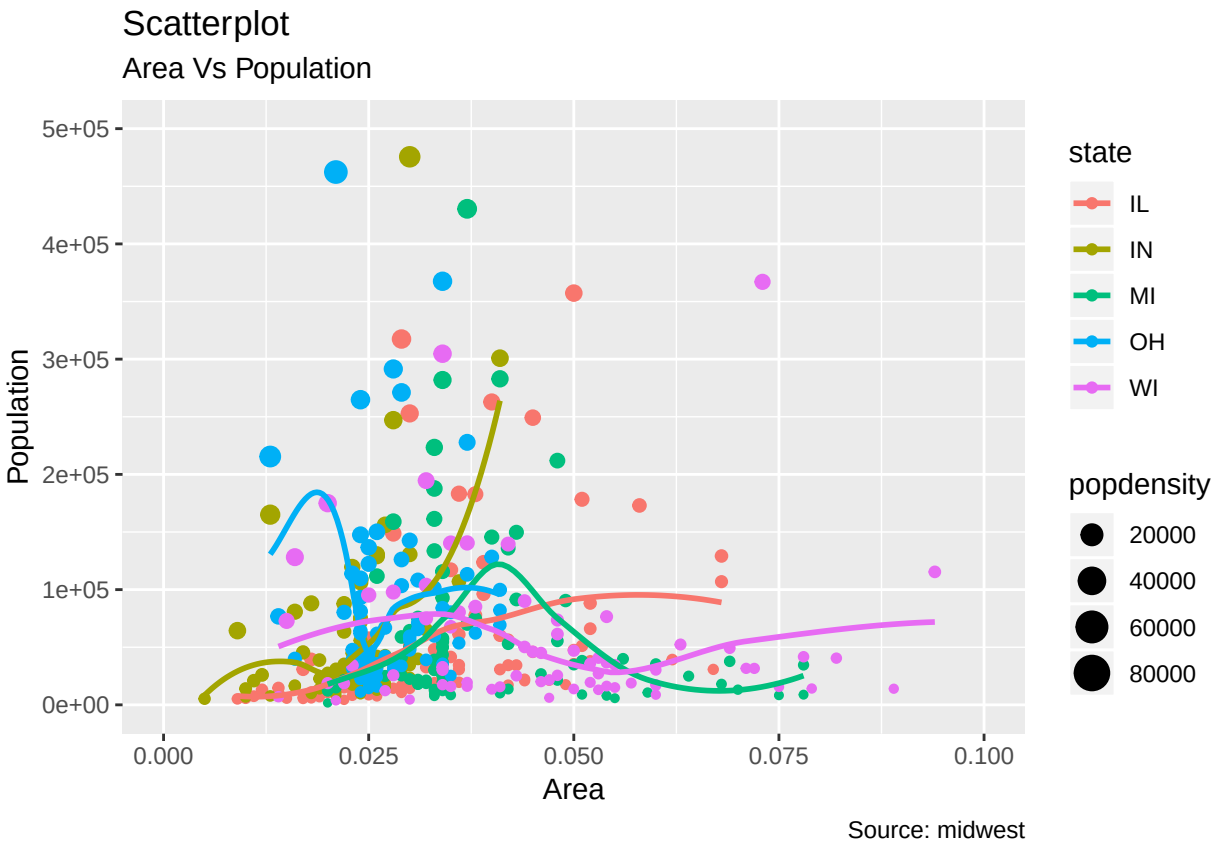
```
## Warning: Removed 177 rows containing non-finite values (stat_ydensity).
```



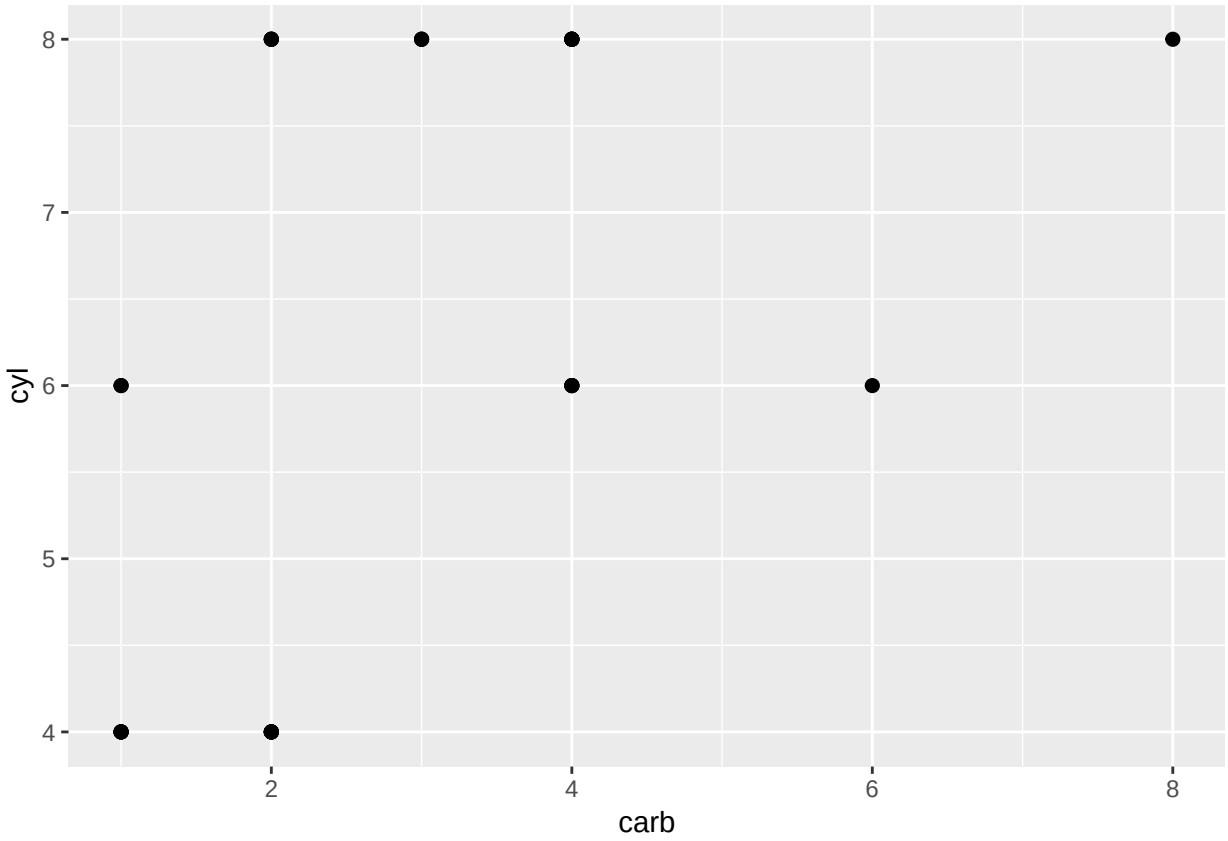

```
##############################################################################
## here's a crazy scatterplot I found online
data("midwest", package = "ggplot2")

## crazy plot!
ggplot(midwest, aes(x = area, y = poptotal, col = state)) +
  geom_point(aes(size=popdensity)) +
  geom_smooth(method="loess", se=F) +
  xlim(c(0, 0.1)) +
  ylim(c(0, 500000)) +
  labs(subtitle="Area Vs Population",
       y="Population",
       x="Area",
       title="Scatterplot",
       caption = "Source: midwest")
```

```
## Warning: Removed 15 rows containing non-finite values (stat_smooth).
```

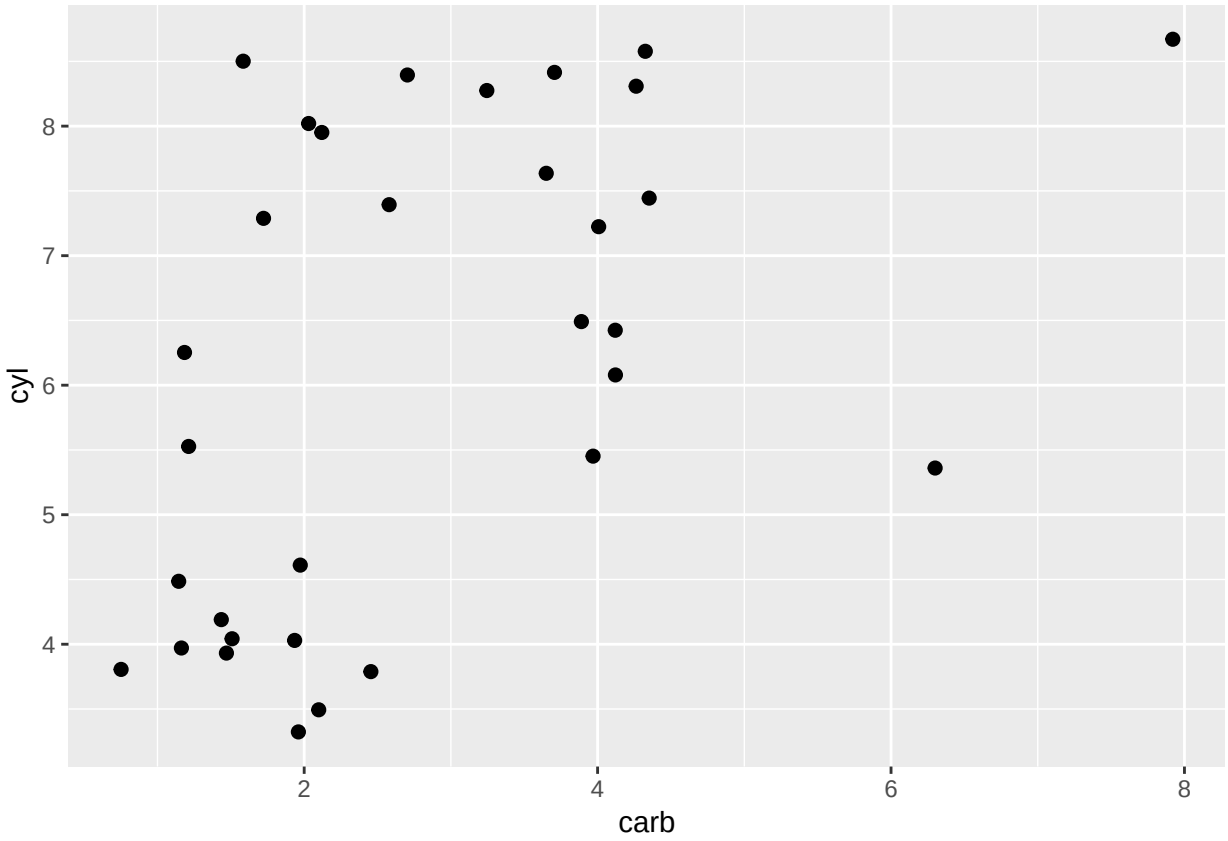
```
## Warning: Removed 15 rows containing missing values (geom_point).
```



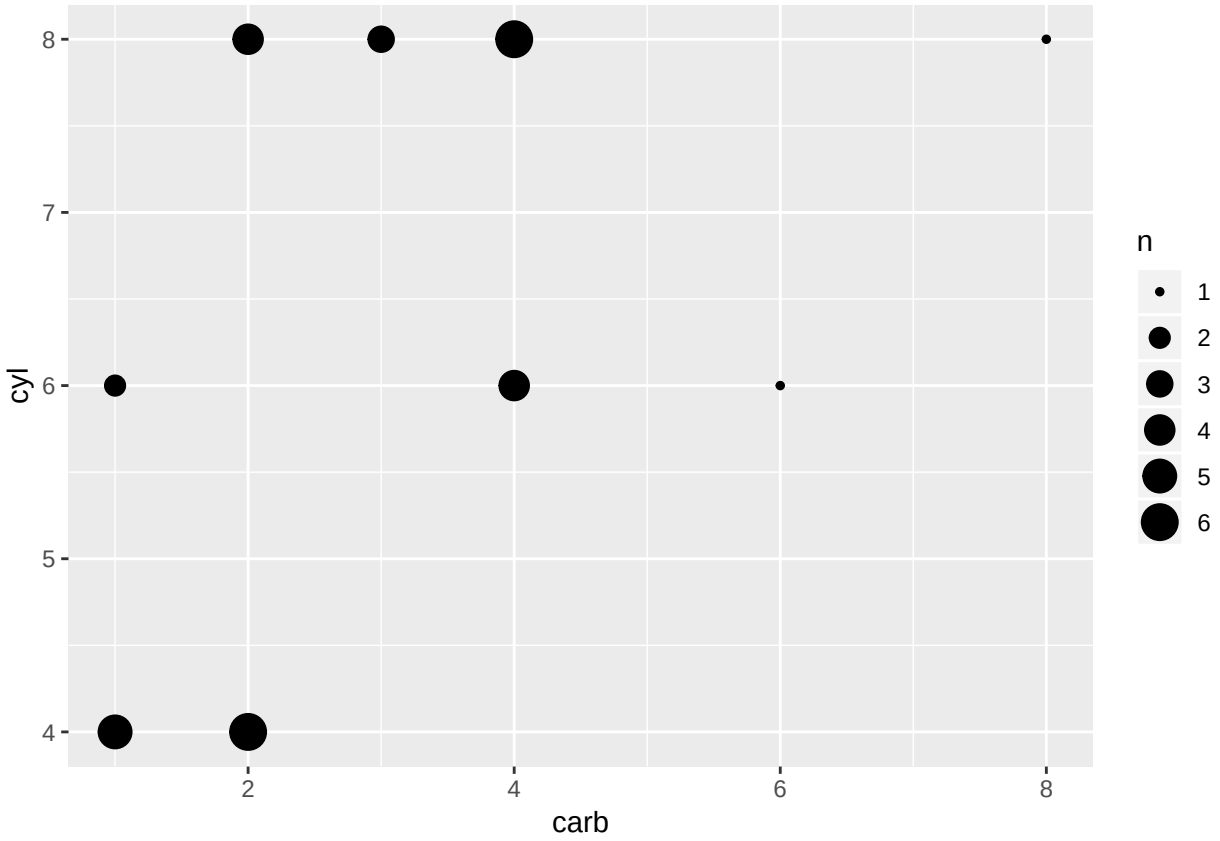

```
##############################################################################
## what if your scatterpoints are overlapping?
data(mpg, package="ggplot2")
mtcars <- mtcars

ggplot(mtcars, aes(x = carb, y = cyl)) +
  geom_point(size = 2) ## :(
```




```
## jitter points around
ggplot(mtcars, aes(x = carb, y = cyl)) +
  geom_jitter(width = .5, size = 2) ## :)
```

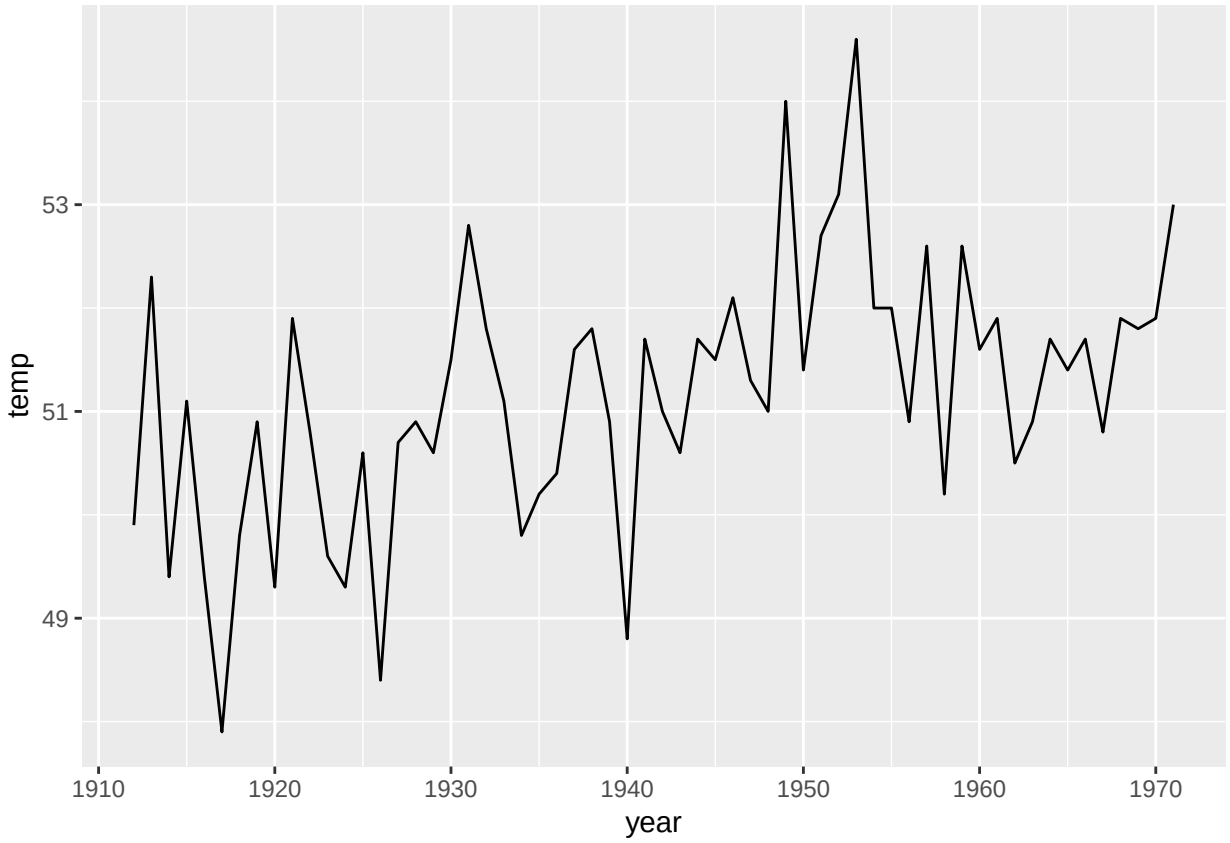



```
## change size to be proportional to count
ggplot(mtcars, aes(x = carb, y = cyl)) +
  geom_count() ## :)
```




```
##############################################################################################
## let's look at some yearly data using the built-in NH temperature data
nh <- as.data.frame(nhtemp)
nh$year <- seq(1912, 1971)
names(nh)[1] <- "temp"

## normal ggplot line graph - how did temp change over the years?
ggplot(nh, aes(x = year, y = temp)) +
  geom_line()
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

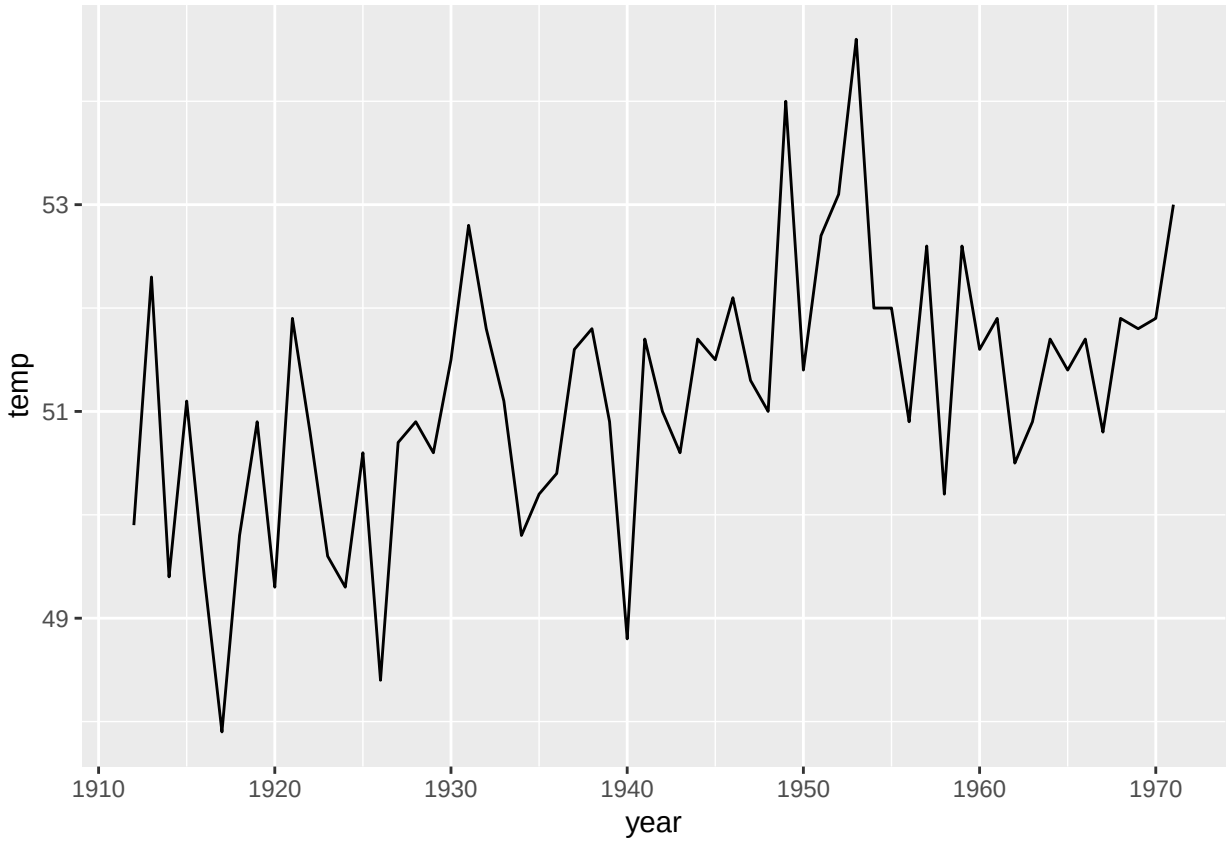


Plotting for presentation

Let's take a plot we made and make it look as good as possible.

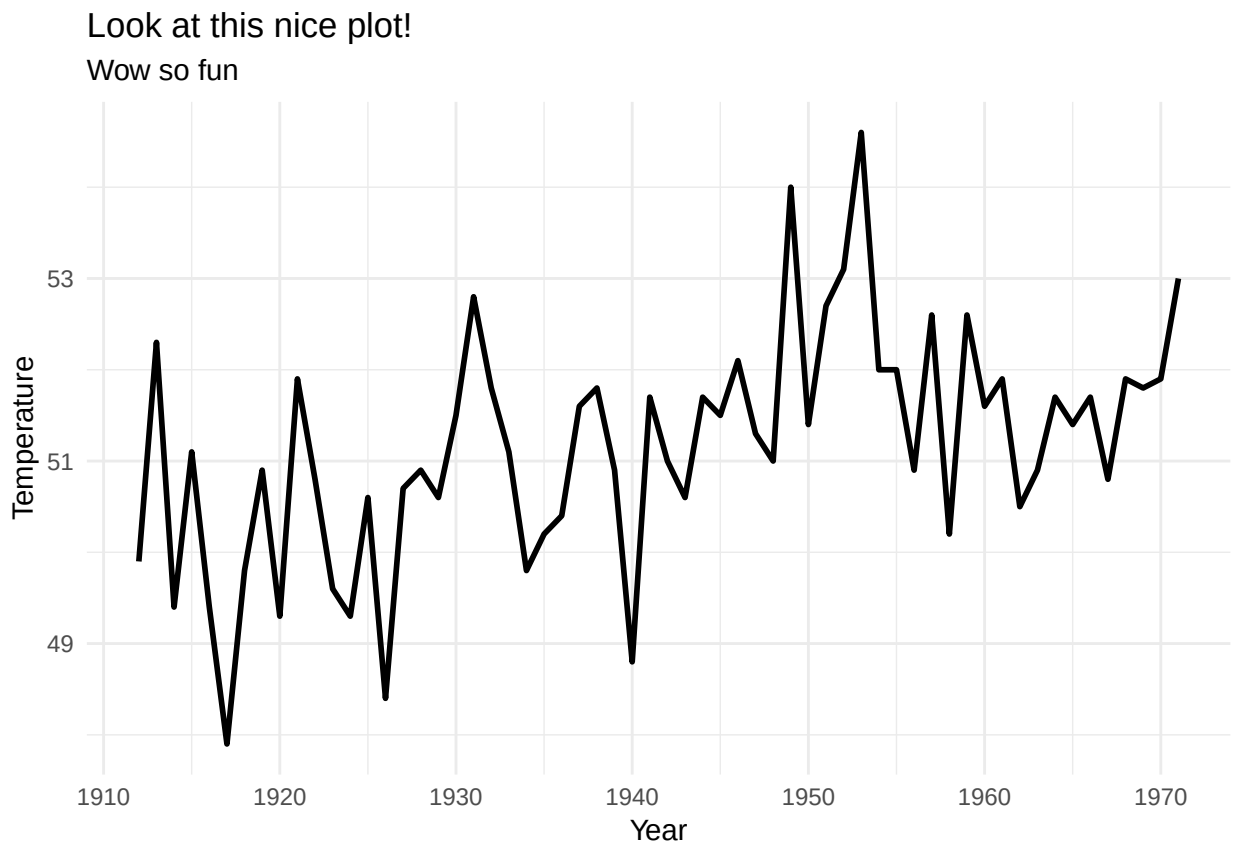
```
## normal ggplot line graph - before our makeover
ggplot(nh, aes(x = year, y = temp)) +
  geom_line()
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




```
## what can we improve?
## I see: background, axis labels, line thickness, title

ggplot(nh, aes(x = year, y = temp)) +
  geom_line(size = 1) +
  theme_minimal() +
  labs(x = "Year", y = "Temperature", title = "Look at this nice plot!", subtitle = "Wow so fun")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




```
## what if we want to save this?

ggsave("NHtemp.png")
```

```
## Saving 6.5 x 4.5 in image
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

```
## the lines look bad, so we have to anti-alias it aka preventing jagged lines
#install.packages('Cairo',,'http://www.rforge.net/')
library(Cairo)
ggsave("NHtemp_smooth.png", type = "cairo")
```

```
## Saving 6.5 x 4.5 in image
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

```
## what if we want to save this with different dimensions?

ggsave("NHtemp_long.png", width = 8, height = 4, type = "cairo")
```

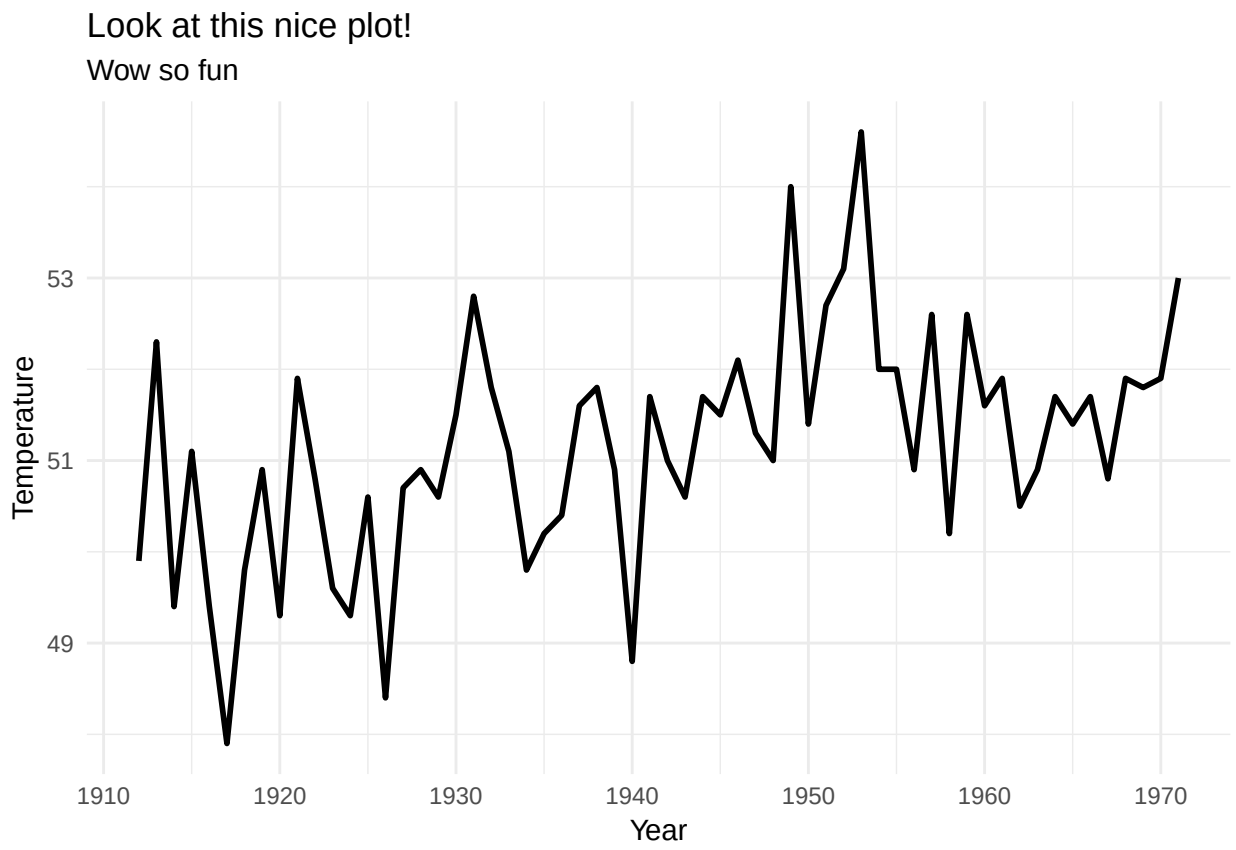
```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```

Fun ending - gganimate!

This is not that practical but is very fun!

```
## let's use our final nhtemp plot
ggplot(nh, aes(x = year, y = temp)) +
  geom_line(size = 1) +
  theme_minimal() +
  labs(x = "Year", y = "Temperature", title = "Look at this nice plot!", subtitle = "Wow so fun")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




```
## animating it!
#install.packages("gganimate")
#install.packages("gifski")
#install.packages("png")

library(gganimate)
library(gifski)
library(png)

anim <-  ggplot(nh, aes(x = year, y = temp)) +
          geom_line(size = 1) +
          theme_minimal() +
          labs(x = "Year", y = "Temperature") +
          transition_reveal(year) ## the only thing we added

#animate(anim, type = "cairo", fps = 6, end_pause = 20) #then we wait
```