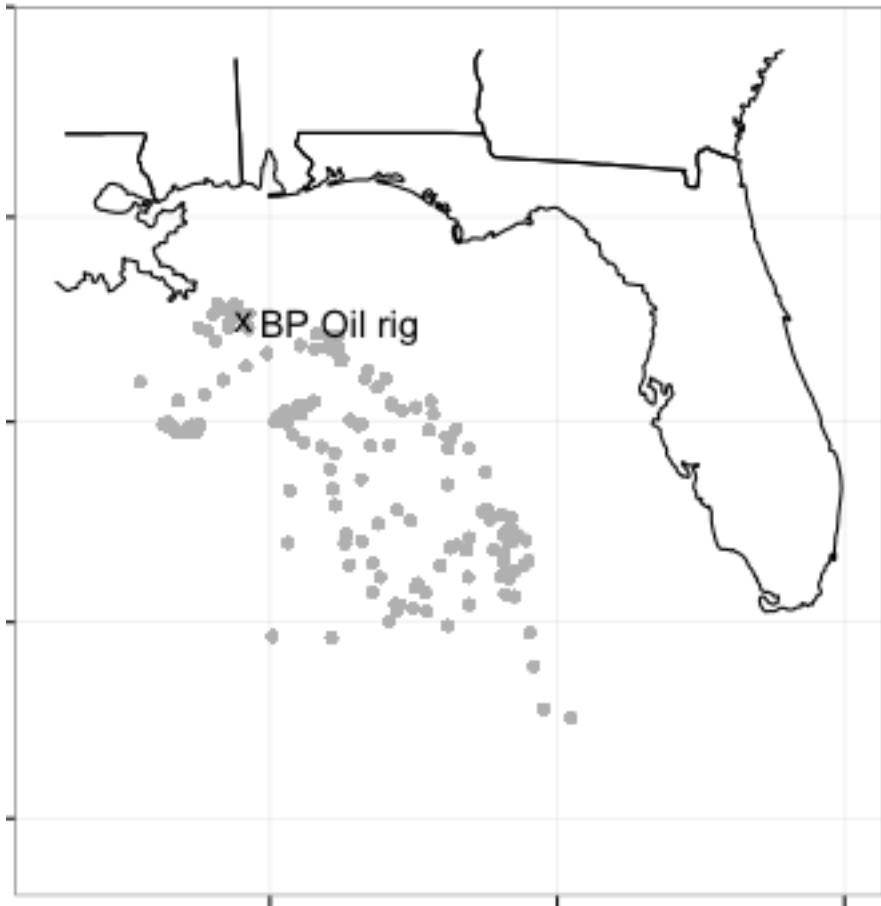


Advanced Plotting

Di Cook, Eric Hare

May 14, 2015



California Dreaming - ASA Travelling Workshop

Back to the Oscars

```
oscars <- read.csv("data/oscars.csv", stringsAsFactors=FALSE)
```

```
acting <- subset(oscars, AwardCategory=="Actor")
```

```
actcountry <- data.frame(table(acting$Country))
```

```
colnames(actcountry)[1] <- "Country"
```

```
head(actcountry)
```

```
##      Country Freq
```

```
## 1 Australia    4
```

```
## 2 Austria     4
```

```
## 3 Belgium     1
```

```
## 4 Cambodia    1
```

```
## 5     Canada     6
## 6     France     4
```

Adding maps

```
library(ggplot2)
library(maps)
library(ggmap)
library(rworldmap)

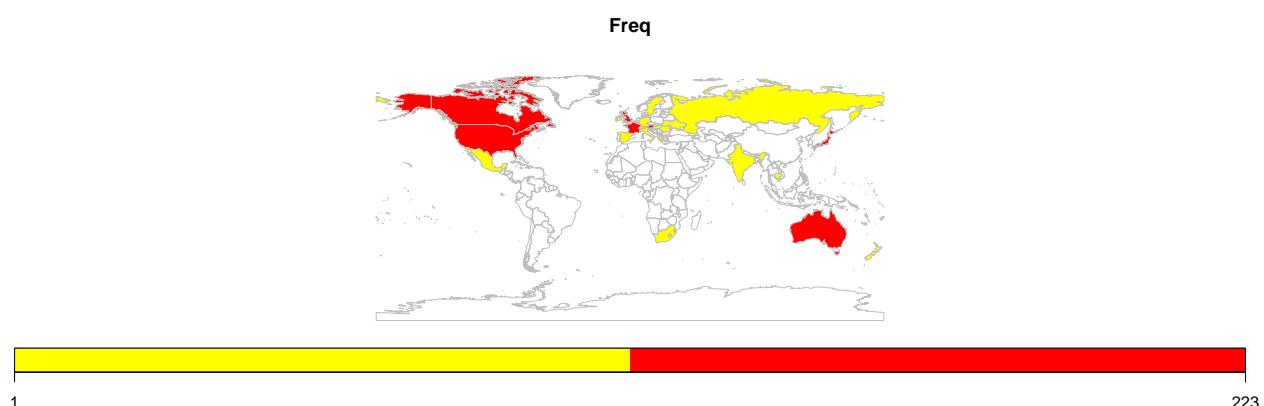
## Loading required package: sp
## ### Welcome to rworldmap ####
## For a short introduction type : vignette('rworldmap')

mc <- joinCountryData2Map(actcountry, joinCode = "NAME",
                           nameJoinColumn = "Country", mapResolution = "low")

## 26 codes from your data successfully matched countries in the map
## 0 codes from your data failed to match with a country code in the map
## 218 codes from the map weren't represented in your data
```

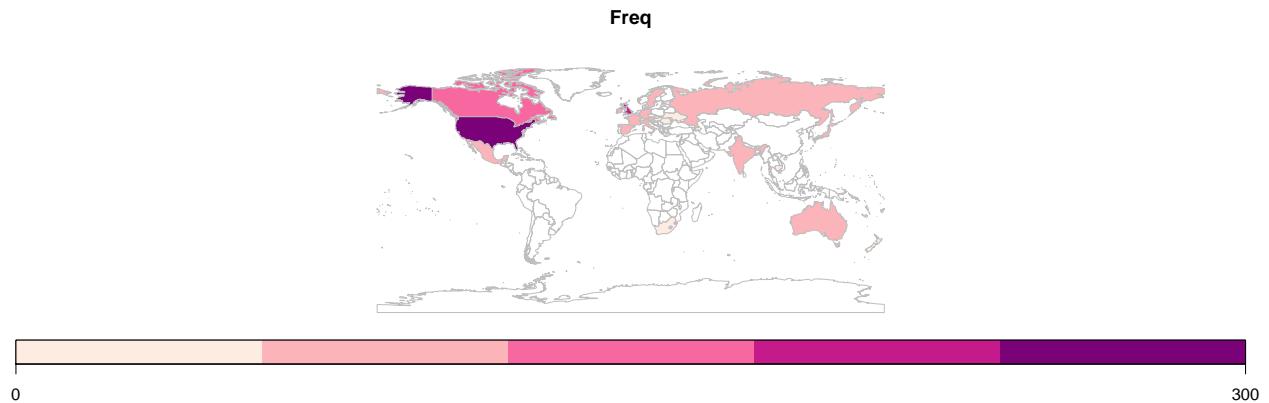
Now draw

```
mapCountryData(mc, nameColumnToPlot="Freq")  
  
## You asked for 7 quantiles, only 2 could be created in quantiles classification
```



And tweak

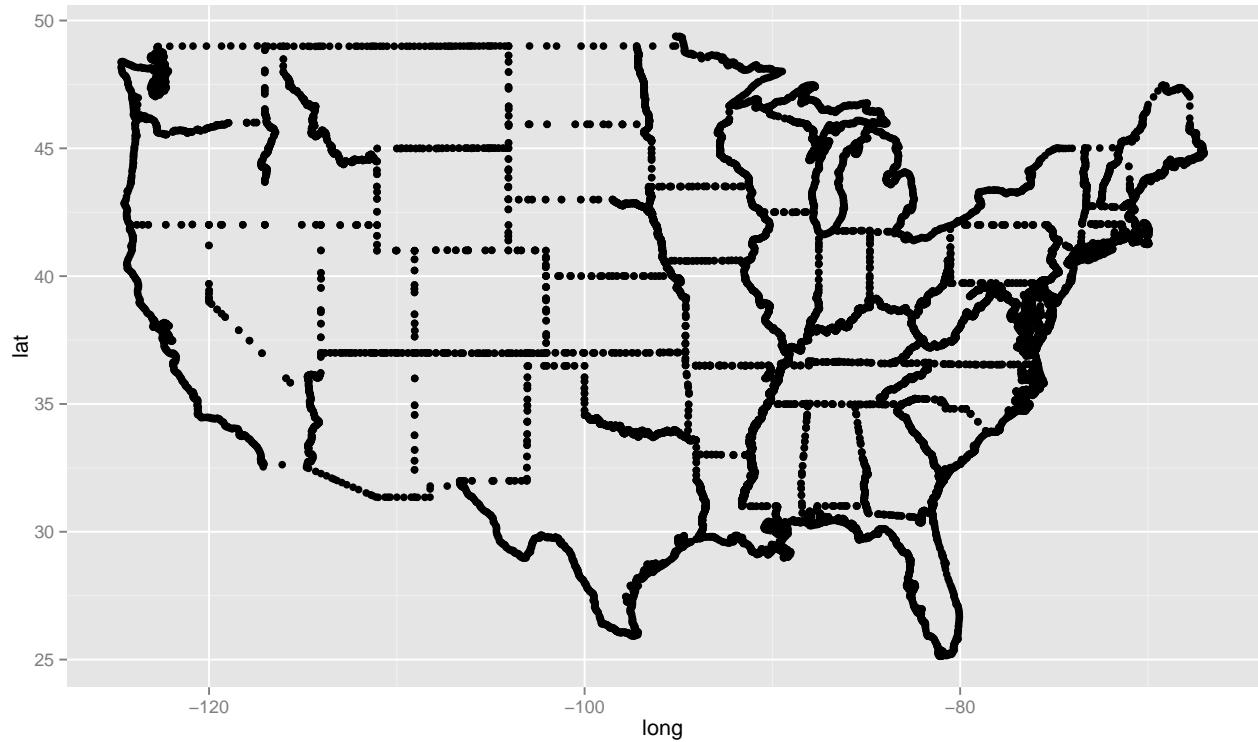
```
library(RColorBrewer)
clrs <- brewer.pal(5, 'RdPu')
mapCountryData(mc, nameColumnToPlot="Freq", catMethod=c(0,1,5, 50, 100, 300),
               colourPalette=clrs)
```



Maps more generally

A map is really just a bunch of latitude longitude points...

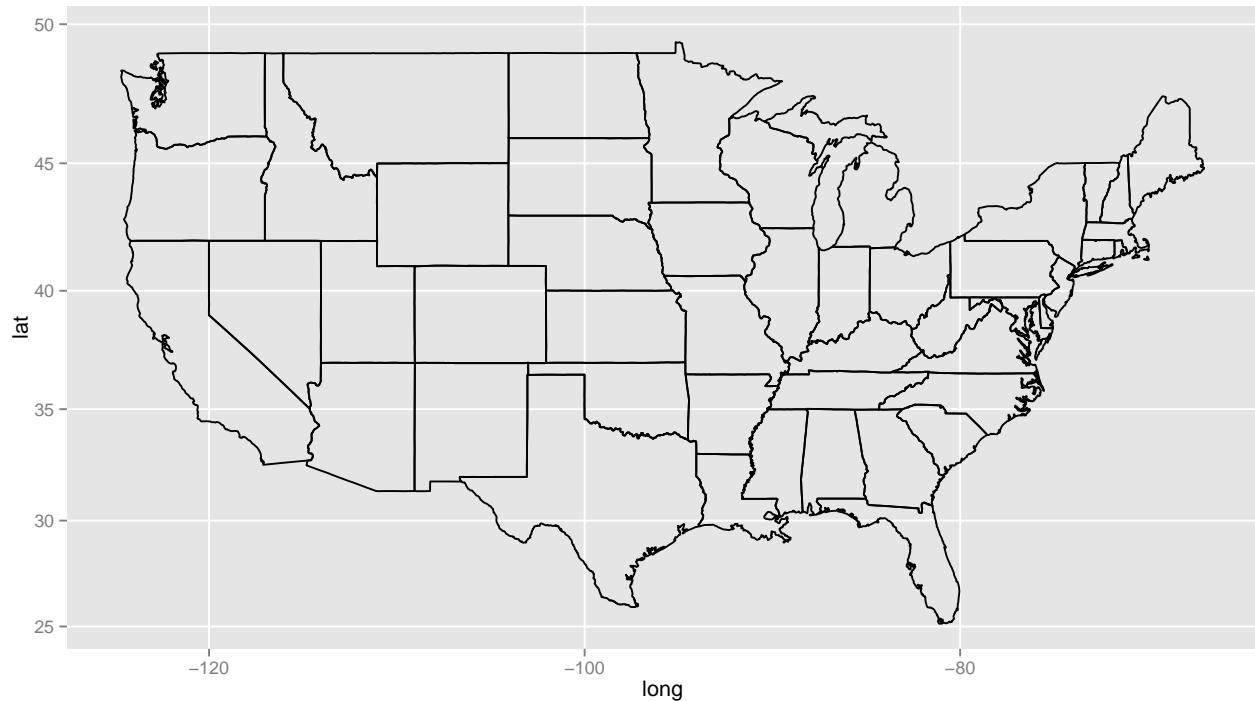
```
qplot(long, lat, geom="point", data=states)
```



Maps more generally

... that are connected with lines in a very specific order.

```
qplot(long, lat, geom="path", data=states, group=group) + coord_map()
```



`coord_map` sets the appropriate lat/long aspect ratio

Basic map data

What needs to be in the data set in order to plot a basic map?

- Need latitude/longitude points for all map boundaries
- Need to know which boundary group all lat/long points belong
- Need to know the order to connect points within each group

Data for building basic state map

Our `states` data has all necessary information

```
states <- map_data("state")
head(states)
```

```
##           long      lat group order region subregion
## 1 -87.46201 30.38968     1     1 alabama      <NA>
```

```

## 2 -87.48493 30.37249      1      2 alabama      <NA>
## 3 -87.52503 30.37249      1      3 alabama      <NA>
## 4 -87.53076 30.33239      1      4 alabama      <NA>
## 5 -87.57087 30.32665      1      5 alabama      <NA>
## 6 -87.58806 30.32665      1      6 alabama      <NA>

```

Incorporating information about states

Want to incorporate additional information into the plot:

- Add other geographic information by adding geometric layers to the plot
- Add non-geographic information by altering the fill color for each state
- Use `geom='polygon'` to treat states as solid shapes to add color
- Incorporate numeric information using color shade or intensity
- Incorporate categorical information using color hue

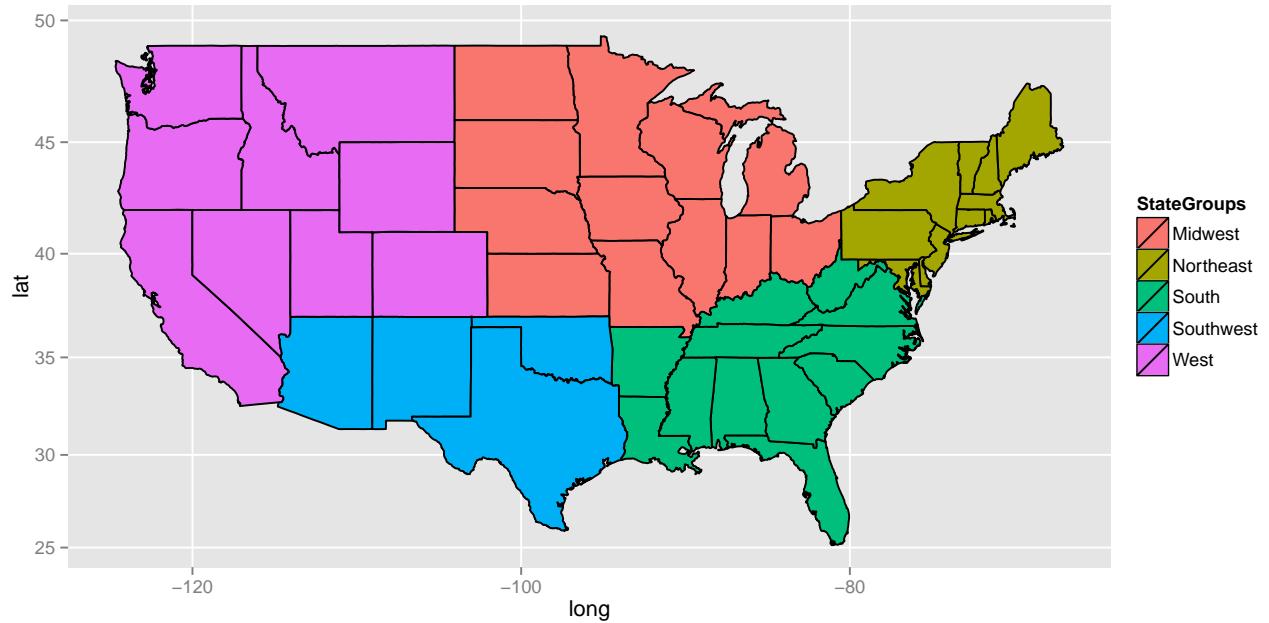
Categorical information using hue

If a categorical variable is assigned as the fill color then `qplot` will assign different hues for each category

```

qplot(long, lat, geom="polygon", data=states.class.map, group=group,
      fill=StateGroups, colour=I("black")) + coord_map()

```



Numerical information using shade and intensity

To show how we can add numerical information to map plots we will use the BRFSS data

- Behavioral Risk Factor Surveillance System
- 2008 telephone survey run by the Center for Disease Control (CDC)
- Ask a variety of questions related to health and wellness
- Cleaned data with state aggregated values posted on website

BRFSS data aggregated by state

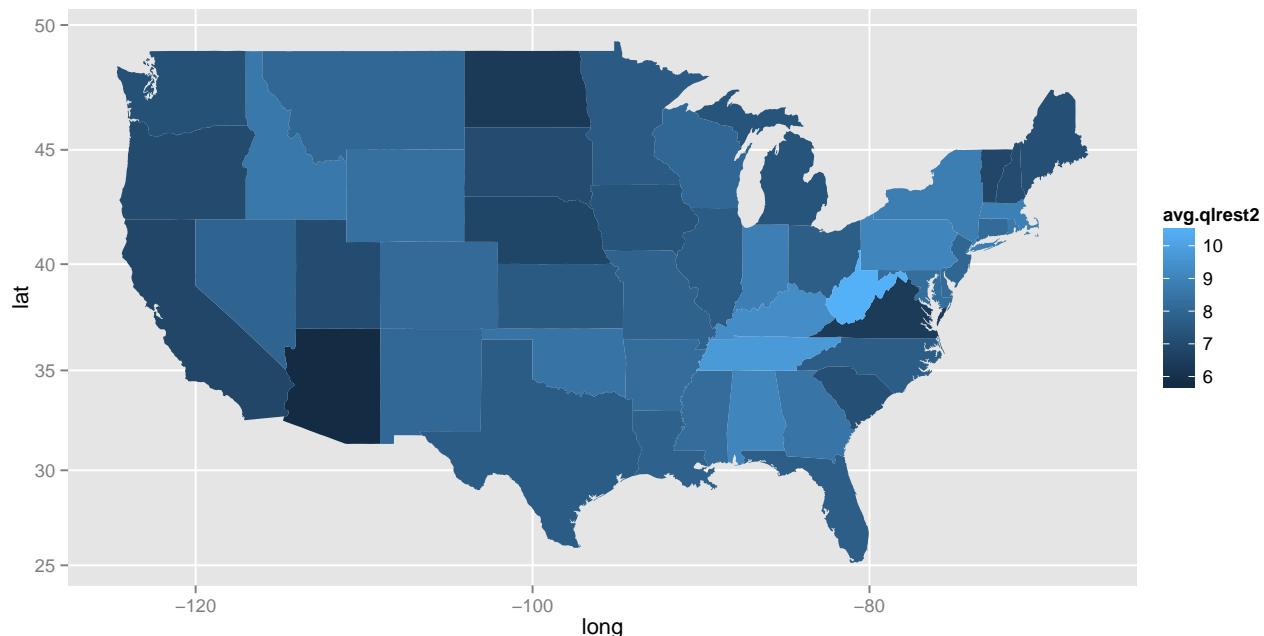
```
head(states.stats)
```

```
##   state.name   avg.wt avg qlrest2   avg.ht   avg.bmi avg.drnk
## 1    alabama 180.7247    9.051282 168.0310 29.00222 2.333333
## 2     alaska 189.2756    8.380952 172.0992 28.90572 2.323529
## 3    arizona 169.6867    5.770492 168.2616 27.04900 2.406897
## 4   arkansas 177.3663    8.226619 168.7958 28.02310 2.312500
## 5  california 170.0464    6.847751 168.1314 27.23330 2.170000
## 6  colorado 167.1702    8.134715 169.6110 26.16552 1.970501
```

Numerical Information Using Shade and Intensity

Average number of days in the last 30 days of insufficient sleep by state

```
states.map <- merge(states, states.stats, by.x="region", by.y="state.name", all.x=T)
qplot(long, lat, geom="polygon", data=states.map, group=group, fill=avg qlrest2) + coord_map()
```



BRFSS Data Aggregated by State and Gender

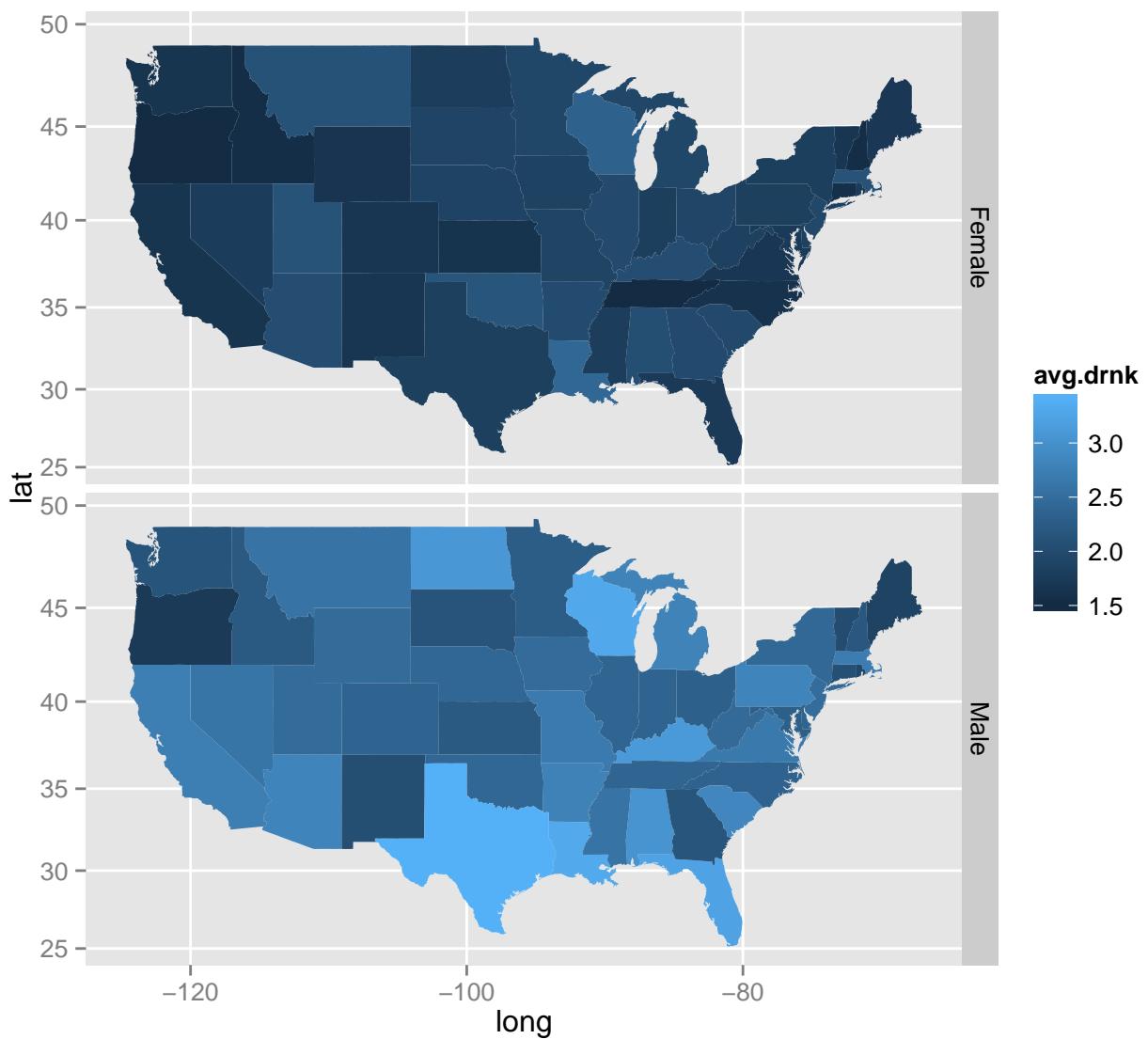
```
states.sex.map <- merge(states, states.sex.stats, by.x="region", by.y="state.name", all.x=T)
head(states.sex.stats)
```

```
##   state.name SEX   avg.wt avg qlrest2   avg.ht avg.bmi avg.drnk     sex
## 1    alabama   1 198.8936    8.648936 177.5729 28.50714 3.033333   Male
## 2    alabama   2 173.0315    9.224771 163.9956 29.21280 2.041667 Female
## 3    alaska     1 203.3919    7.236111 178.3896 28.91494 2.487179   Male
## 4    alaska     2 169.5660    9.907407 163.1296 28.89286 2.103448 Female
## 5   arizona     1 191.3739    5.163793 177.1724 27.63152 2.814286   Male
## 6   arizona     2 156.2054    6.142857 162.7043 26.67683 2.026667 Female
```

Adding Numerical Information

Average number of alcoholic drinks per day by state and gender

```
qplot(long, lat, geom="polygon", data=states.sex.map,
      group=group, fill=avg.drnk) +
      coord_map() + facet_grid(sex ~ .)
```



Your turn

- Use `merge` to combine child healthcare data with maps information
- Then use `qplot` to create a map of child healthcare undercoverage rate by state

Cleaning up your maps

Use ggplot2 options to clean up your map!

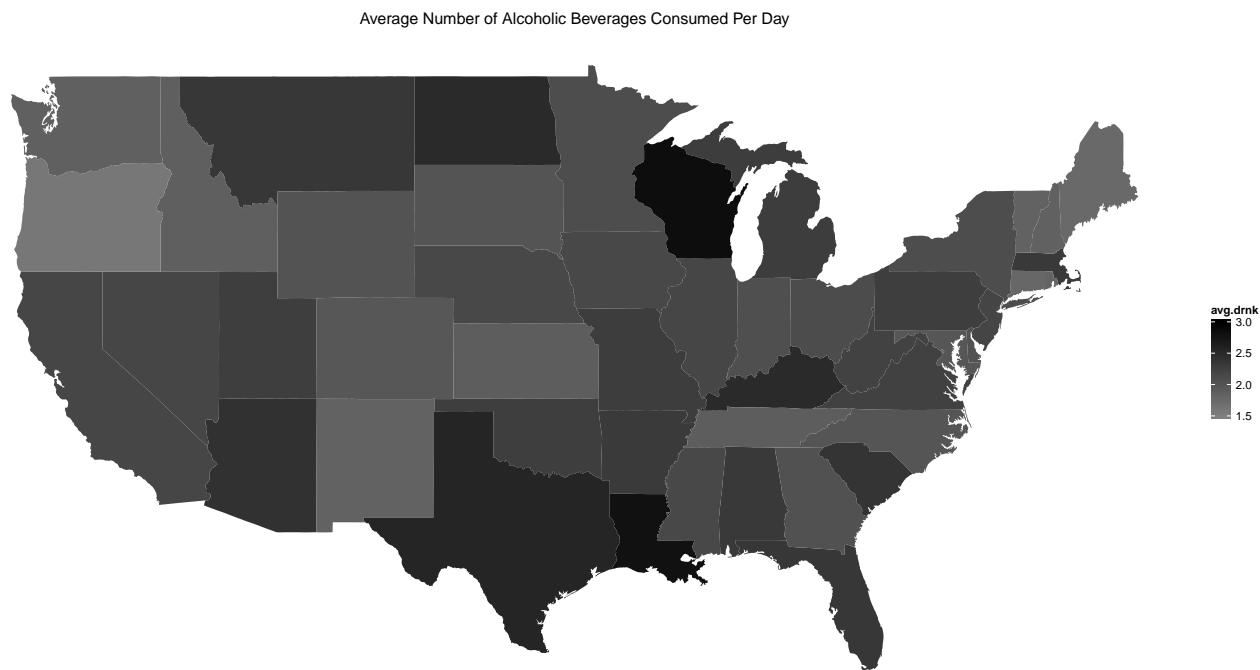
- Adding Titles + `ggtitle(...)`
- Might want a plain white background + `theme_bw()`
- Extremely familiar geography may eliminate need for latitude and longitude axes + `theme(...)`
- Want to customize color gradient + `scale_fill_gradient2(...)`
- Keep aspect ratios correct + `coord_map()`

Cleaning up your maps

```
new_theme_empty <- theme_bw()  
new_theme_empty$line <- element_blank()  
new_theme_empty$rect <- element_blank()  
new_theme_empty$strip.text <- element_blank()  
new_theme_empty$axis.text <- element_blank()  
new_theme_empty$axis.title <- element_blank()  
new_theme_empty$plot.margin <- structure(c(0, 0, -1, -1),  
    unit = "lines", valid.unit = 3L, class = "unit")
```

```
qplot(long, lat, geom="polygon", data=states.map, group=group, fill=avg.drnk) +  
  coord_map() + theme_bw() +  
  scale_fill_gradient2(limits=c(1.5, 3), low="lightgray", high="black") +  
  new_theme_empty +  
  ggtitle("Average Number of Alcoholic Beverages Consumed Per Day")
```

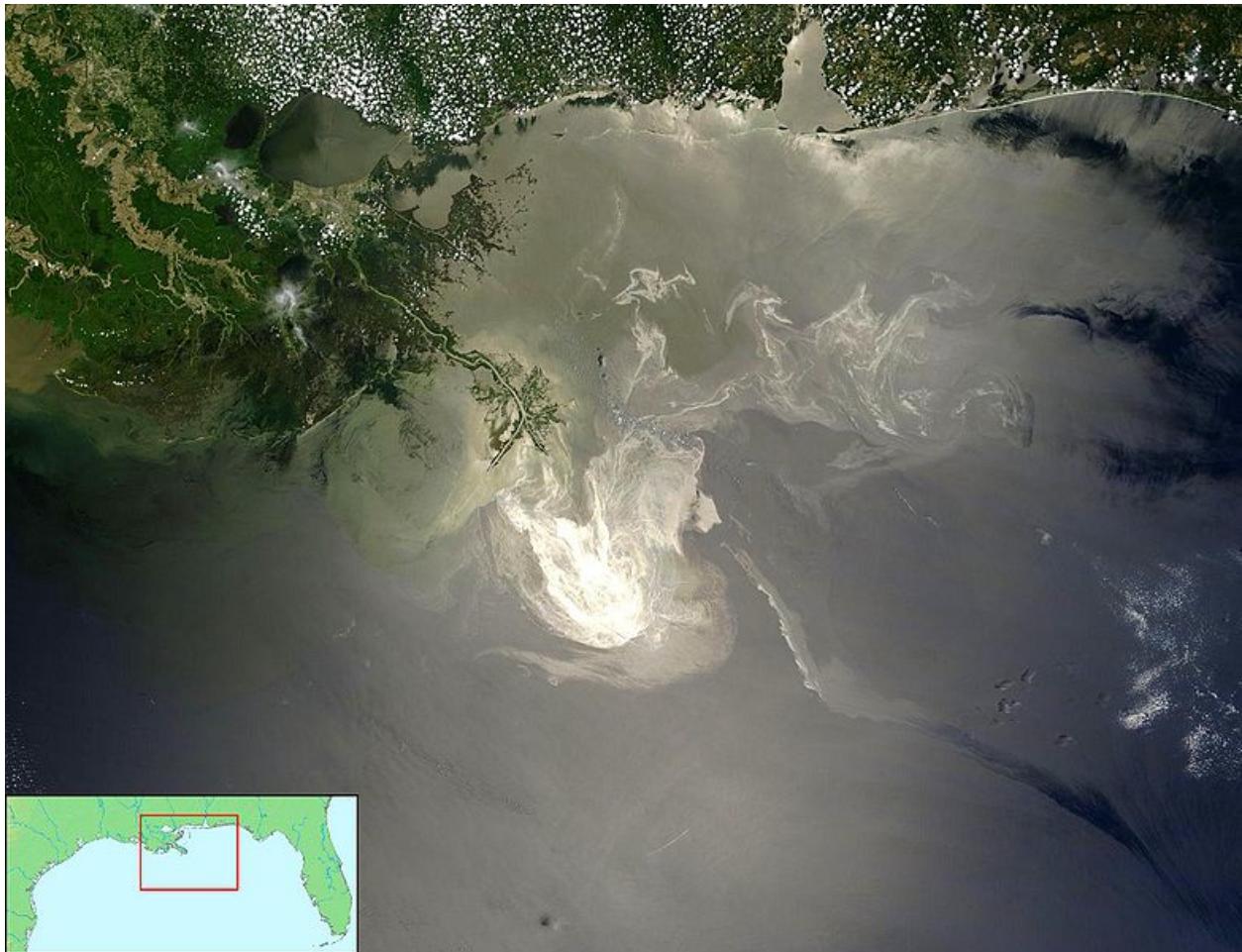
Show it



Your turn

Polish the look of your map of child healthcare undercoverage rate by state!

Putting it together

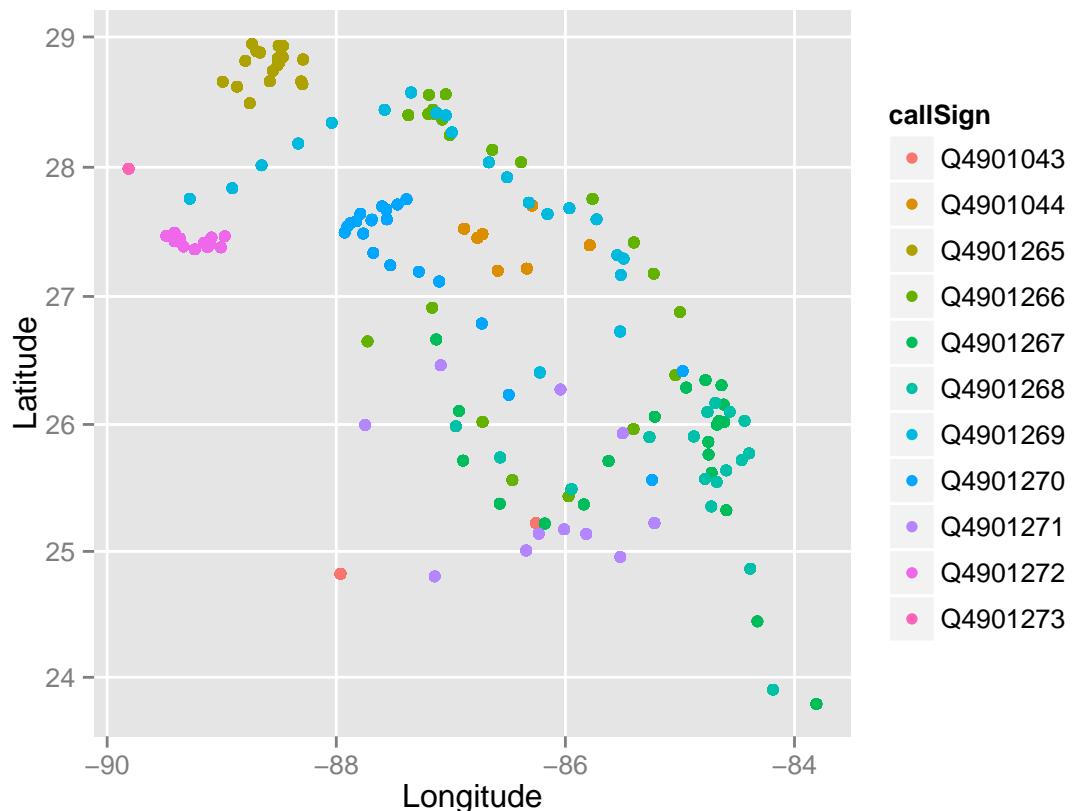


BP Oil Spill May 24 2010 catastrophic environmental disaster in the Gulf. Different measurements provided by NOAA, EPA, US Fish and Wildlife.

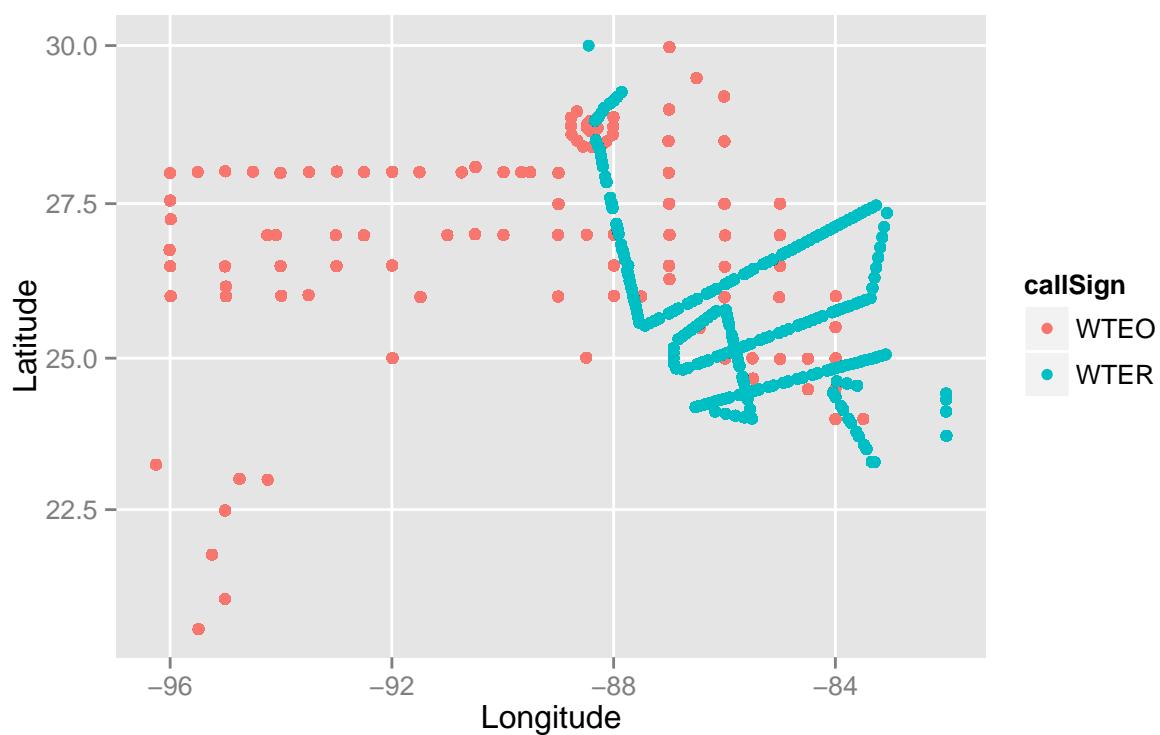
```
load("data/noaa.rdata")
animals <- read.csv("data/animal.csv")
```

Map the data

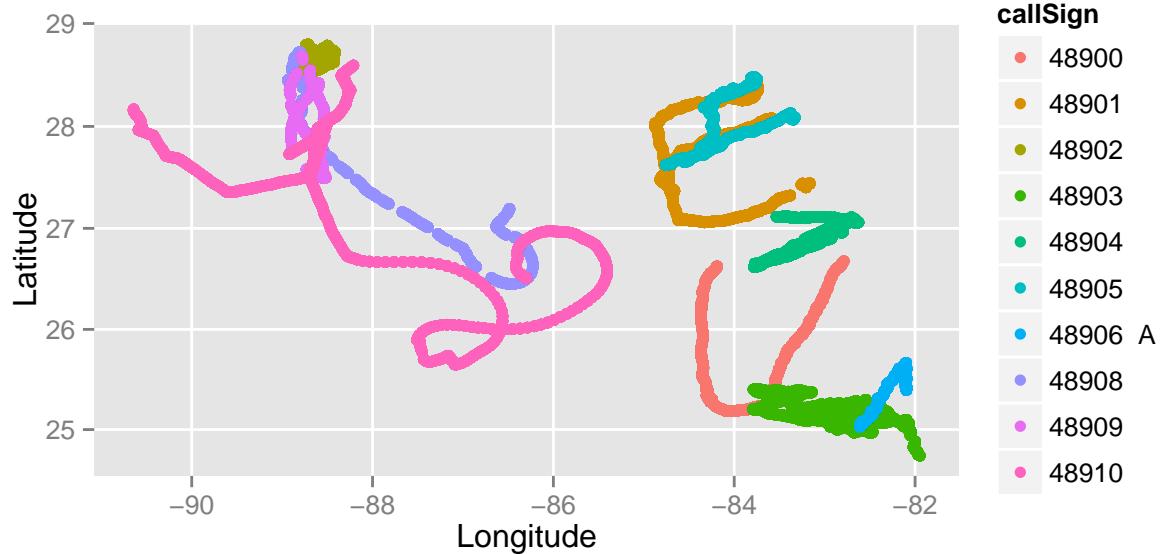
```
qplot(Longitude, Latitude, colour=callSign, data=floating) + coord_map()
```



```
qplot(Longitude, Latitude, colour=callSign, data=boats) + coord_map()
```



```
qplot(Longitude, Latitude, colour=callSign, data=gliders) + coord_map()
```

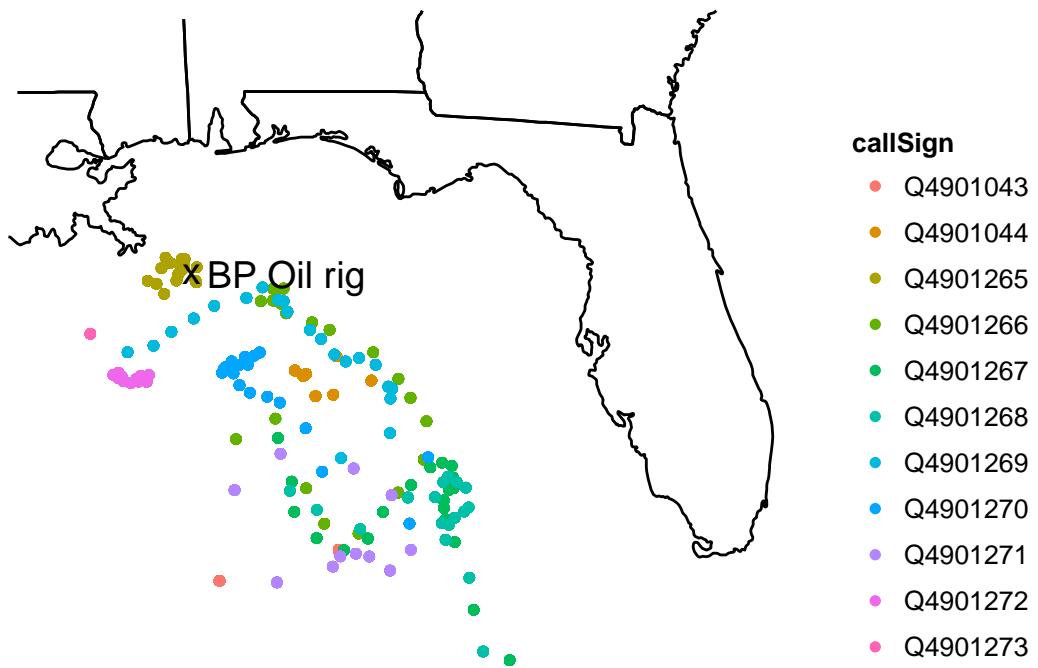


Add a map

```
ggplot() + # plot without a default data set
  geom_path(data=states, aes(x=long, y=lat, group=group)) +
  geom_point(data=floats, aes(x=Longitude, y=Latitude, colour=callSign)) +
  geom_point(aes(x, y), shape="x", size=5, data=rig) +
  geom_text(aes(x, y), label="BP Oil rig", shape="x", size=5, data=rig, hjust = -0.1) +
  xlim(c(-91, -80)) +
  ylim(c(22,32)) + coord_map() + new_theme_empty
```

```
## Warning in loop_apply(n, do.ply): Removed 819 rows containing missing
## values (geom_path).
```

```
## Warning in loop_apply(n, do.ply): Removed 819 rows containing missing
## values (geom_path).
```



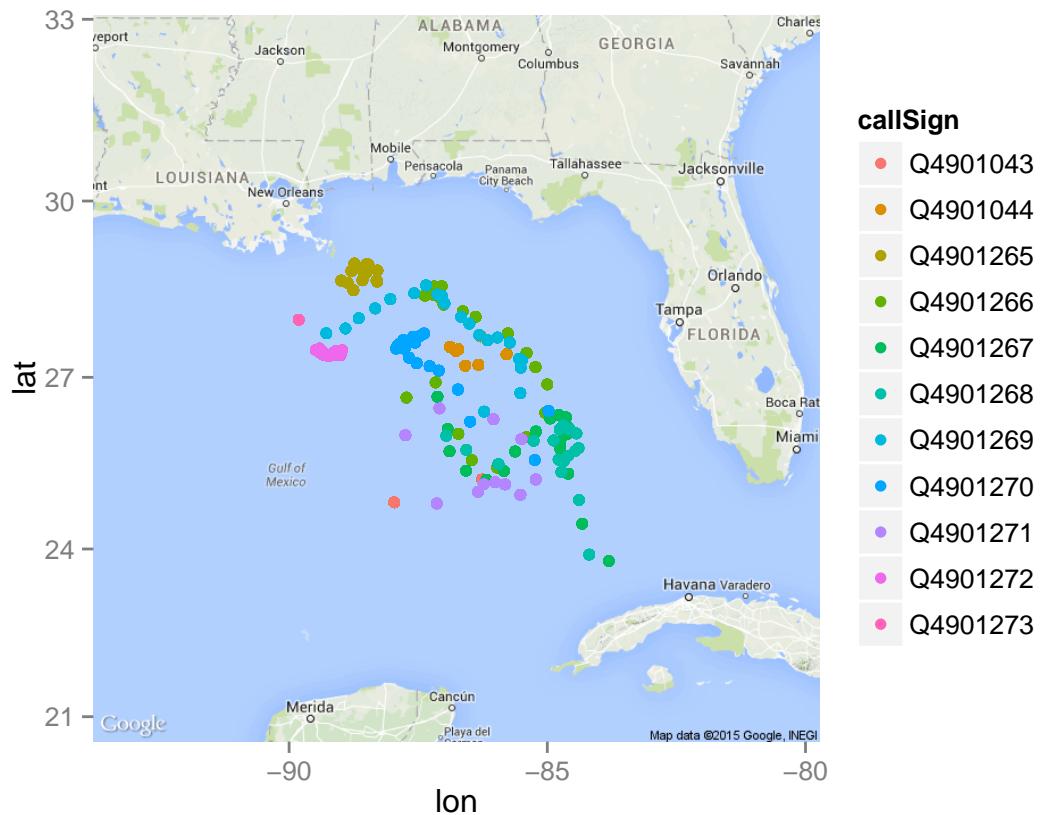
Get fancy

```
gm <- get_googlemap(center = c(lon = -86.77, lat = 26.99), zoom=6)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=26.99,-86.77&zoom=6&size=640x640
```

```
ggmap(gm) + geom_point(data=floats, aes(x=Longitude, y=Latitude, colour=callSign))
```

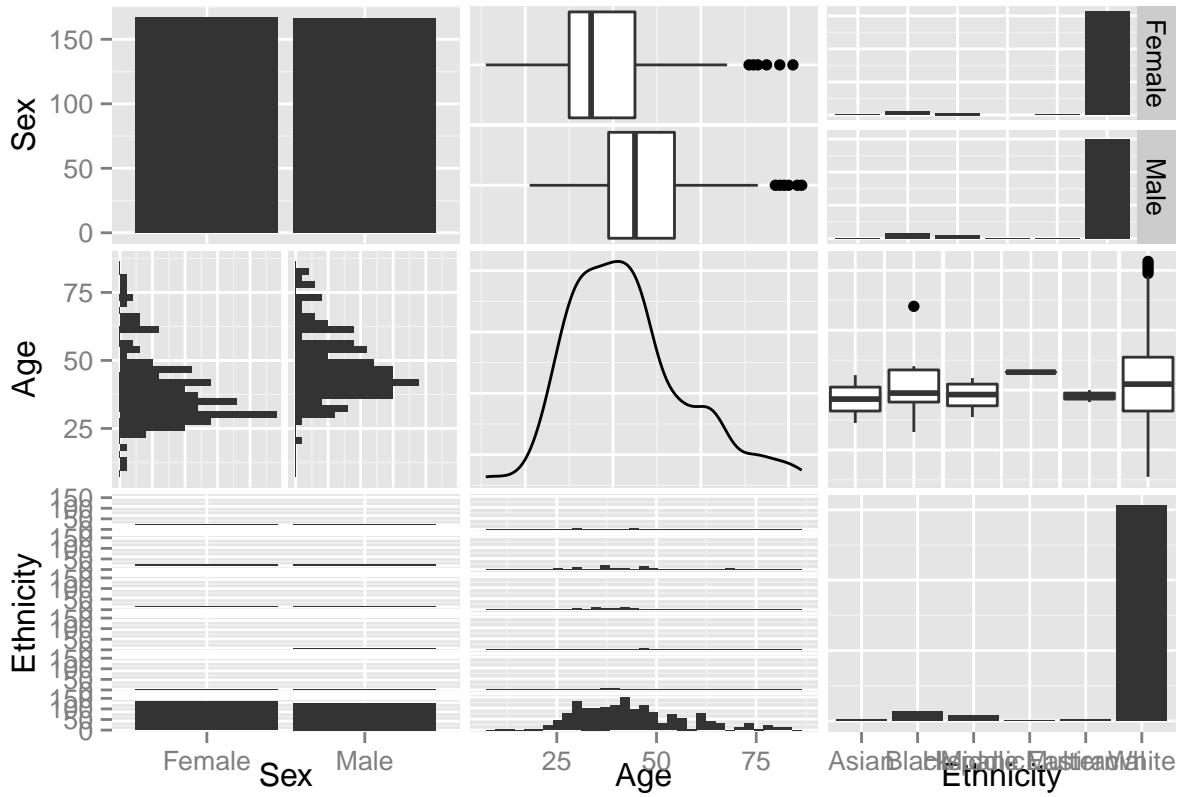
```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=26.99,-86.77&zoom=6&size=640x640
```



Pairs Plot

A scatterplot matrix allows all pairs of numeric variables to be examined, in a manner similar to looking at a correlation matrix. The generalized pairs plot, places appropriate plots of pairs of variables in the cells depending on the type of variable.

```
library(GGally)
ggpairs(acting, columns=c(2,6,8))
```

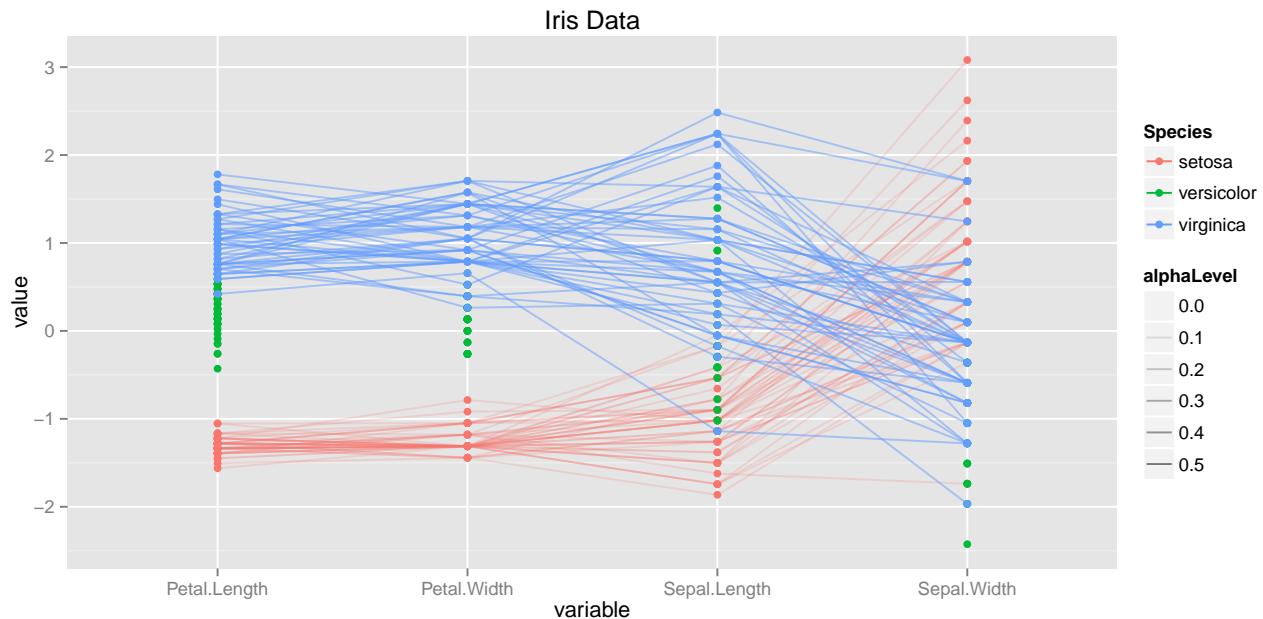


Parallel coordinate plots

```

iris2 <- iris
iris2$alphaLevel <- c("setosa" = 0.2,
  "versicolor" = 0, "virginica" = 0.5)[iris2$Species]
gpd <- ggparcoord(data = iris2,
  columns = 1:4,
  groupColumn = 5, order = "anyClass",
  showPoints = TRUE,
  title = "Iris Data",
  alphaLines = "alphaLevel")

```



Arranging multiple plots on a page

```

acting$Decade <- floor(acting$Year/10)*10
library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:GGally':
##
##     nasa
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

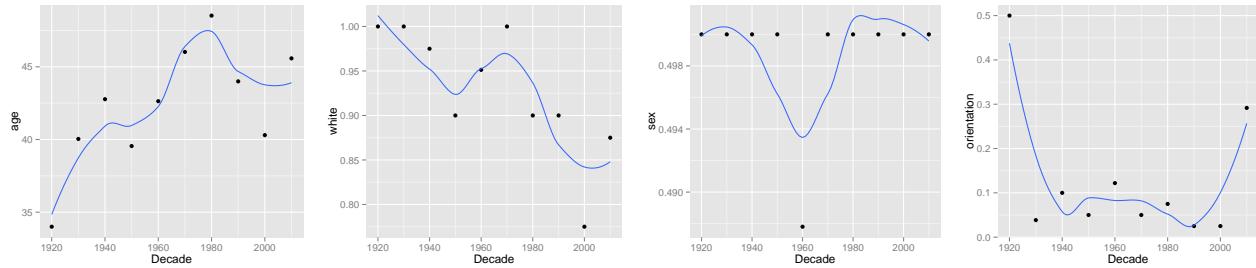
as <- summarise(group_by(acting, Decade), age=mean(Age, na.rm=T),
                 white=length(Ethnicity[Ethnicity=="White"])/length(Ethnicity),
                 sex=length(Sex[Sex=="Male"])/length(Sex),
                 orientation=length(SexualOrientation[SexualOrientation=="Bisexual"])/
                   length(SexualOrientation))

p1 <- qplot(Decade, age, data=as) + geom_smooth(se=F)
p2 <- qplot(Decade, white, data=as) + geom_smooth(se=F)
p3 <- qplot(Decade, sex, data=as) + geom_smooth(se=F)
p4 <- qplot(Decade, orientation, data=as) + geom_smooth(se=F)

```

Plot it

```
library(gridExtra)  
  
## Loading required package: grid  
  
grid.arrange(p1, p2, p3, p4, ncol=4)
```



Summary

- Grammar is good!
- Almost anything is possible
- R Graphics Cookbook by Winston Chang <http://www.cookbook-r.com/Graphs/>
- <http://stackoverflow.com>