DATA 1010
Class Notes
Samuel S. Watson
12 December 2018

Example 1

Make a map of the United States which colors each state according to a column of values stored in a data frame.

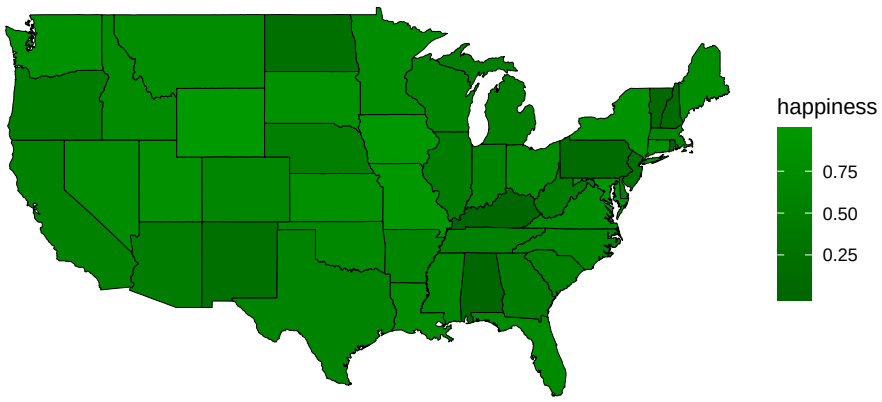
Solution

We pull in the latitude-longitude data per state from the `maps` library, and we populate a data frame called `happiness.df` with made-up data. We `left_join` that data frame with the maps data frame and pipe the result into a `ggplot`. We use the polygon geom to draw the states, we remove the axes with `theme_void()`, and we adjust the colors and set the coordinate system for latitude-longitude data.

```
library(tidyverse)
library(maps)
states <- map_data("state")

statelist <- unique(states$region)
happiness <- runif(length(statelist))
happiness.df <- tibble(region = statelist, happiness = happiness)

left_join(states, happiness.df, by = 'region') %>%
ggplot(aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = happiness), color = 'black', size=0.1) + theme_void() +
  scale_fill_gradient(low = '#006600', high = '#009900') + coord_map()
```



Problem 2

Perform a logistic regression in R using the `caret` package to determine the conditional probability that a child in the `HistData::GaltonFamilies` data frame is female, given their height.

Solution

Note: this solution draws heavily from `rafalab.github.io/dsbook`

We begin by randomly making two groups of records, one for training and one for testing.

```
library(HistData)
library(caret)
test_index <- createDataPartition(GaltonFamilies$childHeight, times = 1, p = 0.05, list = FALSE)

train_set <- GaltonFamilies %>% slice(-test_index)
test_set <- GaltonFamilies %>% slice(test_index)
```

Next we make a plot to show the proportions of children of each height (rounded to the nearest inch) which are female.

```
prop.by.height <- GaltonFamilies %>%
  mutate(height = round(childHeight)) %>%
  group_by(height) %>%
  filter(n() >= 10) %>% # discard sparsely represented heights
  summarize(prop = mean(gender == "female")) %>%
  ggplot(aes(height, prop)) +
  geom_point()
```

To perform a logistic regression, we use a generalized linear model (`glm`) and specify the family as binomial. Note the use of the dot to pipe the data frame from the second line into a specific position in the `glm` call. The expression `y ~ childHeight` is a special type in R called a **formula**, which is used to specify regressors and response variables.

```
glm_fit <- train_set %>%
  mutate(y = as.numeric(gender == "female")) %>%
  glm(y ~ childHeight, data = ., family = "binomial")
```

Finally, we can use the resulting model to make predictions on the test set.

```
p_hat_logit <- predict(glm_fit, newdata = test_set, type = "response")

prop.by.height + geom_line(data=test_set,
                           aes(childHeight,p_hat_logit))
```