Code

- Global.R
  - Load data
- Ui.R - layout
  - shinyUI: sidebarPanel, mainPanel
- **Server.R**

Yi Li, March 13, 2017

Analysis

```
output$analysis1 <- renderPlotly({
  if (length(unique(Select_voting()$rcid))==1){
    df <- vote
    # Select voting data based on rcid and merge with idealpoints data
    df <- subset(df, df$rcid %in% min(Select_voting()$rcid))
    ideal <- subset(idealpoints, idealpoints$session %in% min(Select_voting()$session))
    new.data <- merge(ideal,df,by="ccode")
    new.data$xvar <- new.data[[input$variable]]

    new.data$my_text=paste(new.data$countryname)
    plot_ly(data = subset(new.data, regionnew == 'Latin America'),
            x = ~xvar, y = ~reorder(CountryAbb, xvar),
            type = "scatter", mode = "markers",
            color = ~ordvote, colors = c("#2ecc71", "orange", "red", "grey"),
            text = ~my_text, hoverinfo = "text") %>%
      layout(title = "Latin America", showlegend=F,
             yaxis = list(title=F, autotick=F, tickfont = list(size=9)),
             xaxis = list(title = input$variable))
  }
  else{ggplot()+theme_void()+ggtitle("Select the vote title")}
})
```

"If … else …" is useful!

Map

"ploy_geo" uses color bar (Choropleth maps) to show the color of the variable with different values. So I set "yes" as the lowest color value - green, set "no" as the highest color value - red, and set the other two "abstain and DNV" in the middle.

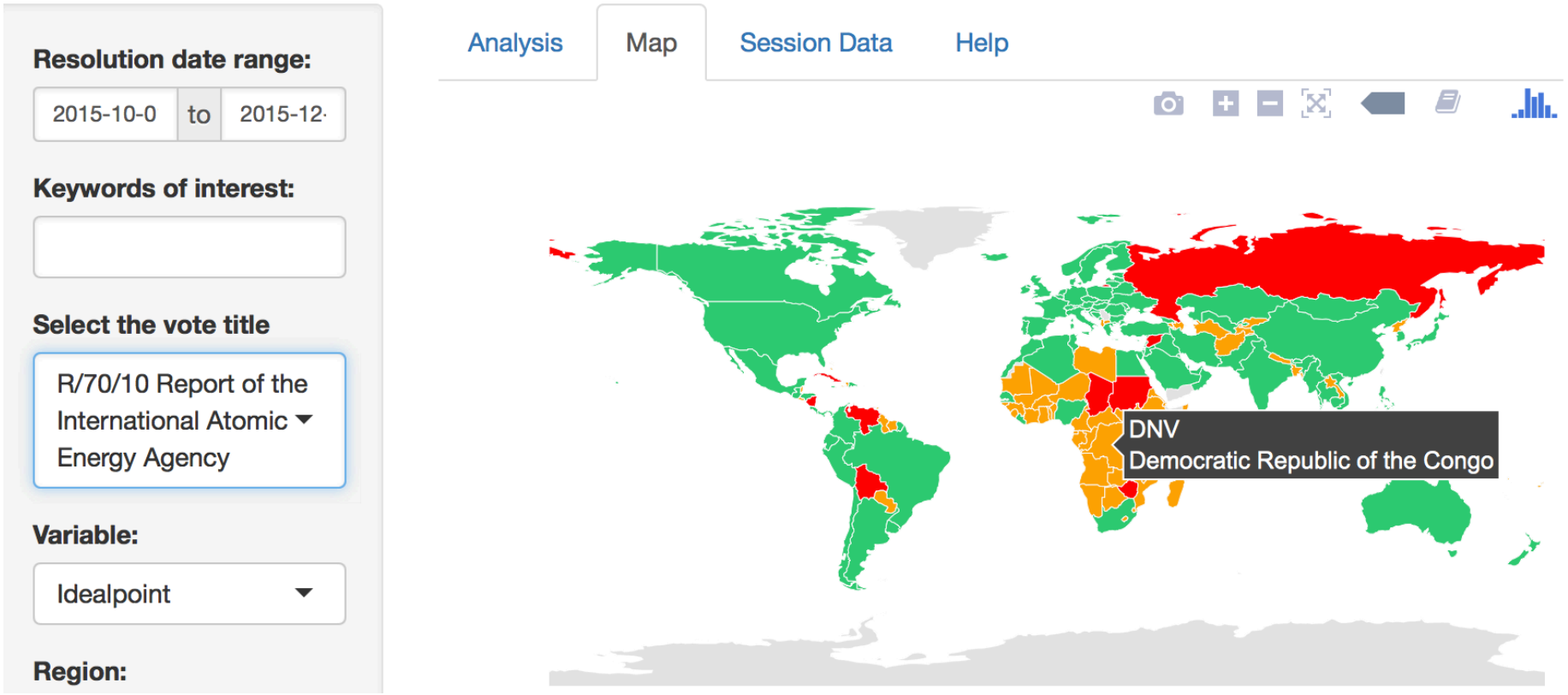
- If there are four vote situations: "yes, no, abstain, DNV", the color is right. (abstain - orange, DNV - grey.)
- If there are two vote situations: "yes, no", the color is also right.

```
mydf$hover <- with(mydf, paste(ordvote,"<br>",Country))
mydf$vote[mydf$Vote=="Absent"] <- 1.5

# light grey boundaries
l <- list(color = toRGB("white"), width = 0.5)

# specify map projection/options
g <- list(
  showframe = F,
  showcoastlines = T,
  coastlinecolor = toRGB('white'),
  showland = T,
  landcolor = toRGB("grey90"),
  projection = list(type = 'azequalarea')
)

p <- plot_geo(mydf,locationmode = 'country names') %>%
  add_trace(
    z = ~vote, color = ~vote,
    colors = c("#2ecc71","grey","orange", "grey","red"),
    text = ~hover, hoverinfo = "text",
    locations = ~Country, marker = list(line = l),
    showscale = F
  ) %>%    #colorbar() %>%
  layout(geo = g)   #title = 'vote',
p
```



But if there are three vote situations: "yes, no, DNV", the color of "DNV" becomes orange.

Prediction Analysis

1. Use the data of a selected year to build a model. For example, choose the variable "Data of Year" (data.year) as 2014.

```
oprobit <- polr(as.factor(Vote) ~ xvar1, data = subset(new.data,
new.data$year == input$data.year & new.data$regionnew ==
input$region), Hess = T, model = TRUE,  method = "probit")
```

2. Use the full dataset to do prediction.

```
probs <- predict(oprobit, new.data, type = "probs")
```

3. Plot the data of the year one wants to predict. For example, choose the variable "Prediction of Year" (pred.year) as 2015.

```
plot_ly (data = subset(new.data, new.data$year == input$pred.year &
new.data$regionnew == input$region), x = ~probs.yes,
y=~reorder(CountryAbb, probs.yes), color = ~ordvote)
```

As I use the full dataset to do prediction in the step2, we can choose any year to make the plot. If we choose pred.year = data.year = 2014, then the training dataset and the validation dataset are the same. Usually, we choose pred.year > data.year to test the model.

Other Challenges

- When I used the package "coefplot", there is a conflict.

"coefplot" is created based on an old version of "ggplot2", since the "ggplot2" is updated, we cannot use "coefplot" as usual. I tried to install an old version of "ggplot2" to solve the problem, but then there is another conflict of "plotly". So instead, I used another package "arm" to do the coefplot.

- If there are only two observed choices, I used the probit model as following:

```
opglm(as.factor(Vote) ~ xvar1, data = subset(new.data,
new.data$year robit <- == input$data.year & new.data$regionnew ==
input$region), family = binomial(link = "probit"))
```

Then I set "probs.no = 1 - probs.yes" without considering "probs.abstain". In this case, I suppose there is always at least one country vote yes. Here is the code:

```
probs <- predict(oprobit, new.data, type = "response")
new.data$probs.yes <- probs
new.data$prob.no <- 1 - probs
```