

Assignment 6: Map and Interactive Graphs

Anshita Thakkar

First, I ran in the dataset

```
marketing_campaign = read.csv("/Users/anshitathakkar/Documents/Visual  
analytics/Week 8_files/Assignment 6/marketing_campaign.csv")  
marketing_campaign
```

The screenshot shows the RStudio interface with the code editor at the top containing the command to read the CSV file. Below it is a data frame viewer titled "Description: df [2,216 x 27]". The data frame contains 2,216 rows and 27 columns. The columns are: ID, Year_Birth, Education, Marital_Status, Income, Kidhome, Teenhome, Dt_Customer, Recency, and MntWines. The data includes various demographic and transactional information. At the bottom of the viewer, there is a navigation bar with buttons for "Previous" and "Next" and page numbers 1 through 100.

| ID | Year_Birth | Education | Marital_Status | Income | Kidhome | Teenhome | Dt_Customer | Recency | MntWines |
|------|------------|------------|----------------|--------|---------|----------|-------------|---------|----------|
| 5524 | 1957 | Graduation | Single | 58138 | 0 | 0 | 9/4/2012 | 58 | 635 |
| 2174 | 1954 | Graduation | Single | 46344 | 1 | 1 | 3/8/2014 | 38 | 11 |
| 4141 | 1965 | Graduation | Together | 71613 | 0 | 0 | 8/21/2013 | 26 | 426 |
| 6182 | 1984 | Graduation | Together | 26646 | 1 | 0 | 2/10/2014 | 26 | 11 |
| 5324 | 1981 | PhD | Married | 58293 | 1 | 0 | 1/19/2014 | 94 | 173 |
| 7446 | 1967 | Master | Together | 62513 | 0 | 1 | 9/9/2013 | 16 | 520 |
| 965 | 1971 | Graduation | Divorced | 55635 | 0 | 1 | 11/13/2012 | 34 | 235 |
| 6177 | 1985 | PhD | Married | 33454 | 1 | 0 | 5/8/2013 | 32 | 76 |
| 4855 | 1974 | PhD | Together | 30351 | 1 | 0 | 6/6/2013 | 19 | 14 |
| 5899 | 1950 | PhD | Together | 5648 | 1 | 1 | 3/13/2014 | 68 | 28 |

Then I ran the packages

```
library(ggplot2)  
library(dplyr)  
install.packages("usmap")  
library(usmap)
```

I loaded “statepop” data

```
statespop = data("statepop")  
statepop
```

The screenshot shows the RStudio interface with the code editor at the top containing the command to load the statepop data. Below it is a data frame viewer titled "A tibble: 51 x 4". The data frame contains 51 rows and 4 columns: fips, abbr, full, and pop_2015. The data lists the US states with their abbreviations and populations. At the bottom of the viewer, there is a navigation bar with buttons for "Previous" and "Next" and page numbers 1 through 6.

| fips | abbr | full | pop_2015 |
|------|------|----------------------|----------|
| 01 | AL | Alabama | 4858979 |
| 02 | AK | Alaska | 738432 |
| 04 | AZ | Arizona | 6828065 |
| 05 | AR | Arkansas | 2978204 |
| 06 | CA | California | 39144818 |
| 08 | CO | Colorado | 5456574 |
| 09 | CT | Connecticut | 3590886 |
| 10 | DE | Delaware | 945934 |
| 11 | DC | District of Columbia | 672228 |
| 12 | FL | Florida | 20271272 |

I created a US map showing state population. Prior to this, I merged the data

```

statepop$region <- tolower(statepop$full)

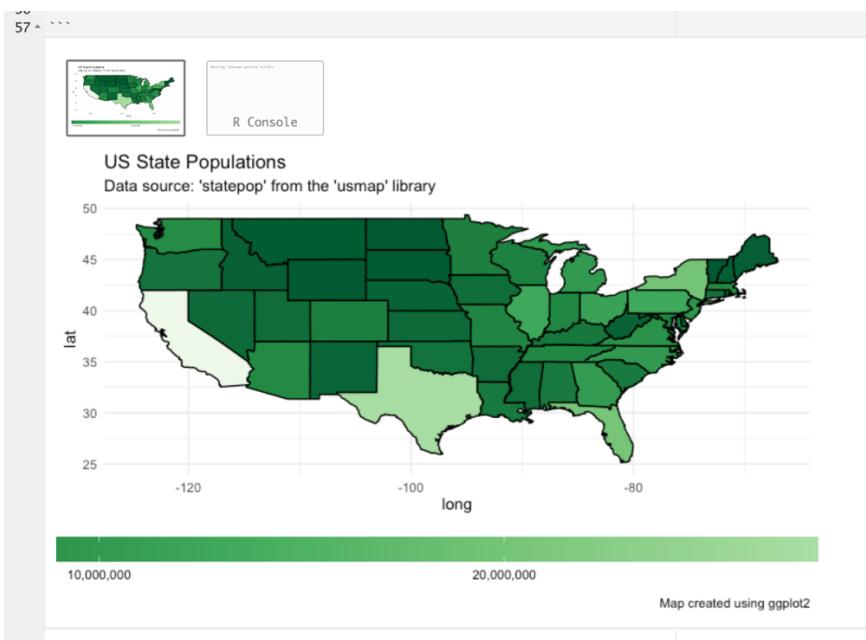
states_map <- map_data("state")

population_map <- merge(states_map, statepop, by.x = "region", by.y = "region")

gg <- ggplot(population_map, aes(x = long, y = lat, group = group, fill = pop_2015)) +
  geom_polygon(color = "black") +
  scale_fill_distiller(palette = "Viridis", name = "Population", labels = scales::comma) +
  labs(title = "US State Populations",
       subtitle = "Data source: 'statepop' from the 'usmap' library",
       caption = "Map created using ggplot2") +
  theme_minimal() +
  theme(legend.position = "bottom")

gg + theme(legend.key.width = unit(3, "in"))

```



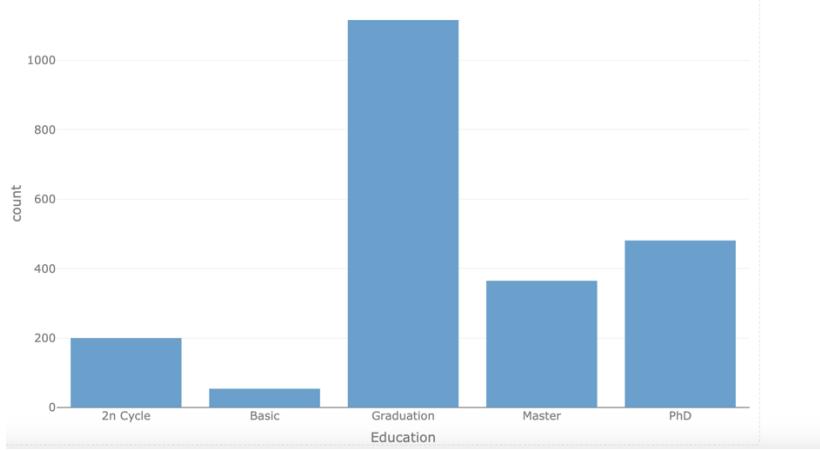
I checked the structure: str(marketing_campaign)

```
'data.frame': 2216 obs. of 27 variables:
$ ID          : int  5524 2174 4141 6182 5324 7446 965 6177 4855 5899 ...
$ Year_Birth   : int  1957 1954 1965 1984 1981 1967 1971 1985 1974 1950 ...
$ Education    : chr "Graduation" "Graduation" "Graduation" "Graduation" ...
$ Marital_Status: chr "Single" "Single" "Together" "Together" ...
$ Income       : int  58138 46344 71613 26646 58293 62513 55635 33454 30351 5648 ...
$ Kidhome      : int  0 1 0 1 1 0 1 1 1 ...
$ Teenhome     : int  0 1 0 0 0 1 1 0 1 ...
$ Dt_Customer  : chr "9/4/2012" "3/8/2014" "8/21/2013" "2/10/2014" ...
$ Recency      : int  58 38 26 26 94 16 34 32 19 68 ...
$ MntWines     : int  635 11 426 11 173 520 235 76 14 28 ...
$ MntFruits    : int  88 1 49 4 43 42 65 10 0 0 ...
$ MntMeatProducts: int  546 6 127 20 118 98 164 56 24 6 ...
$ MntFishProducts: int  172 2 111 10 46 0 50 3 3 1 ...
$ MntSweetProducts: int  88 1 21 3 27 42 49 1 3 1 ...
$ MntGoldProds  : int  88 6 42 5 15 14 27 23 2 13 ...
$ NumDealsPurchases: int  3 2 1 2 5 2 4 2 1 1 ...
$ NumWebPurchases: int  8 1 8 2 5 6 7 4 3 1 ...
$ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
$ NumStorePurchases: int  4 2 10 4 6 10 7 4 2 0 ...
$ NumWebVisitsMonth: int  7 5 4 6 5 6 6 8 9 20 ...
$ AcceptedCmp3  : int  0 0 0 0 0 0 0 0 0 1 ...
$ AcceptedCmp4  : int  0 0 0 0 0 0 0 0 0 0 ...
$ AcceptedCmp5  : int  0 0 0 0 0 0 0 0 0 0 ...
$ AcceptedCmp1  : int  0 0 0 0 0 0 0 0 0 0 ...
$ AcceptedCmp2  : int  0 0 0 0 0 0 0 0 0 0 ...
$ Complain      : int  0 0 0 0 0 0 0 0 0 0 ...
$ Response      : int  1 0 0 0 0 0 0 0 1 0 ...
```

I created interactive bar plots using categorical variables.

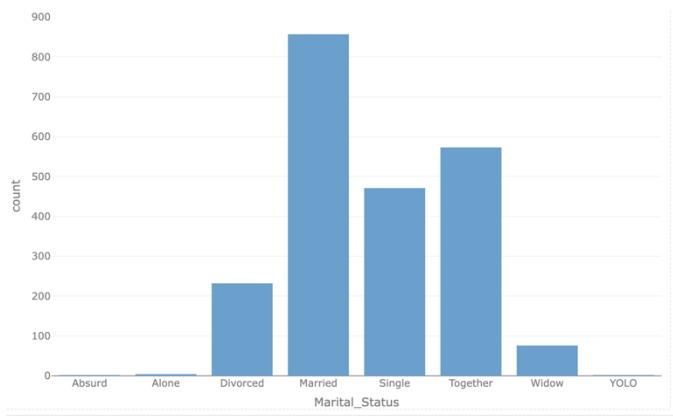
```
library(plotly)
plot1 <- marketing_campaign %>%
  group_by(Education) %>%
  summarise(count = n()) %>%
  plot_ly(x = ~Education, y = ~count, type = "bar", name = "Education Levels")
```

plot1



```
plot3 <- marketing_campaign %>%
  group_by(Marital_Status) %>%
  summarise(count = n()) %>%
  plot_ly(x = ~Marital_Status, y = ~count, type = "bar", name = "Marital Status")
```

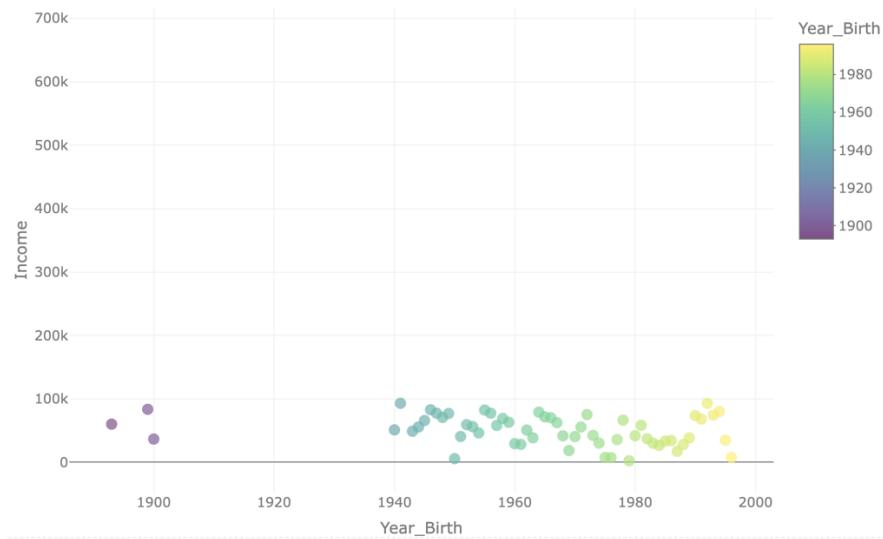
plot3



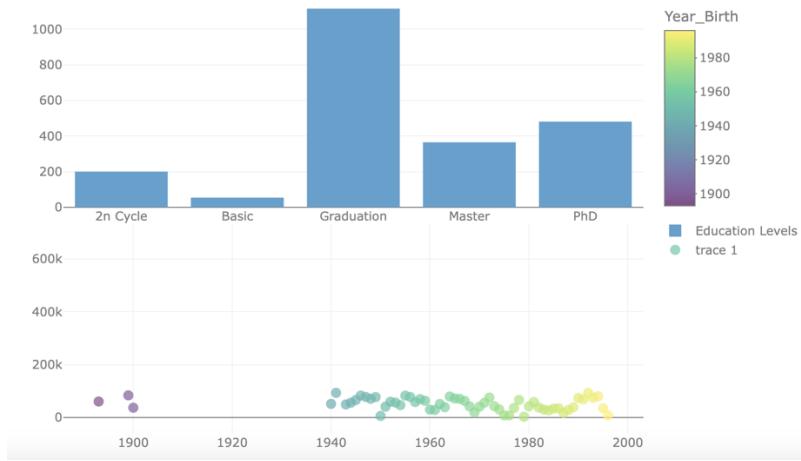
I created an interactive scatterplot using numeric variables

```
plot2 <- marketing_campaign %>%
  plot_ly(x = ~Year_Birth, y = ~Income, type = "scatter", mode = "markers", color =
  ~Year_Birth, size = ~Income,
  marker = list(size = 10, opacity = 0.7, colorscale = "Viridis"), ids = ~Year_Birth)
```

plot2



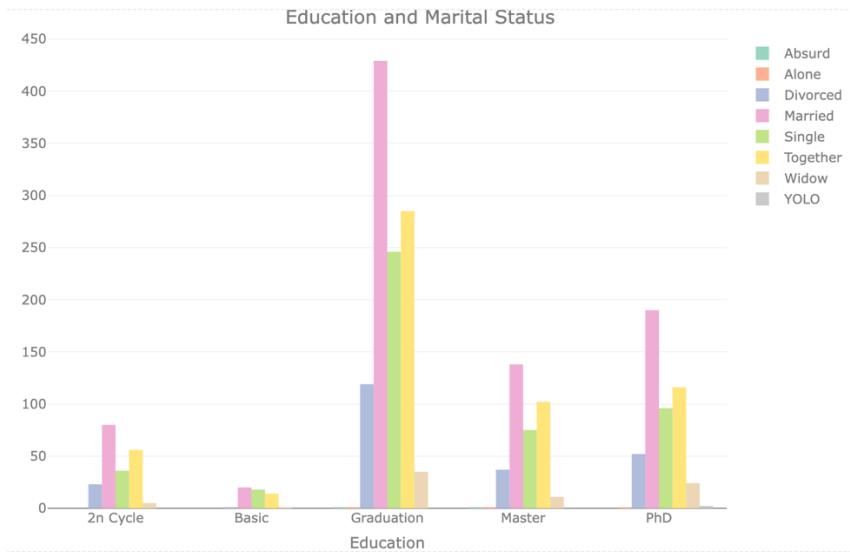
I presented the two plots together



```
subplot(plot1, plot2, nrows = 2)
```

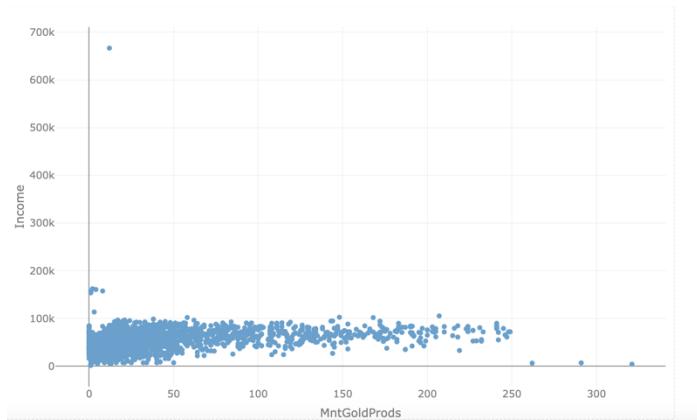
I created a grouped bar plot by Education and Marital Status.

```
library(plotly)
multi_title <- layout(plot_ly(marketing_campaign, x = ~Education, color =
~Marital_Status), title = "Education and Marital Status")
multi_title
```

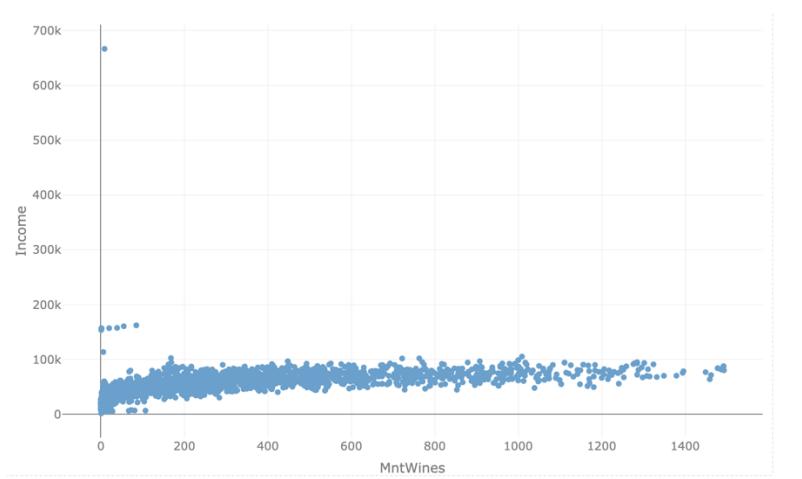


I created scatter plots using numerical variables

```
scatter1 <- plot_ly(marketing_campaign, x = ~MntGoldProds, y = ~Income)
scatter1
```



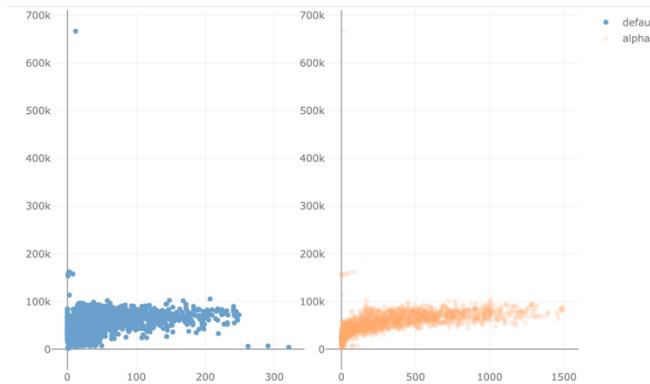
```
scatter2 <- plot_ly(marketing_campaign, x = ~MntWines, y = ~Income)  
scatter2
```



I grouped the scatter plots together

```
scatter3 <- subplot(  
  plot_ly(marketing_campaign, x = ~MntGoldProds, y = ~Income, name = "default"),  
  plot_ly(marketing_campaign, x = ~MntWines, y = ~Income) %>% add_markers(alpha  
= 0.2, name = "alpha")  
)
```

```
scatter3
```

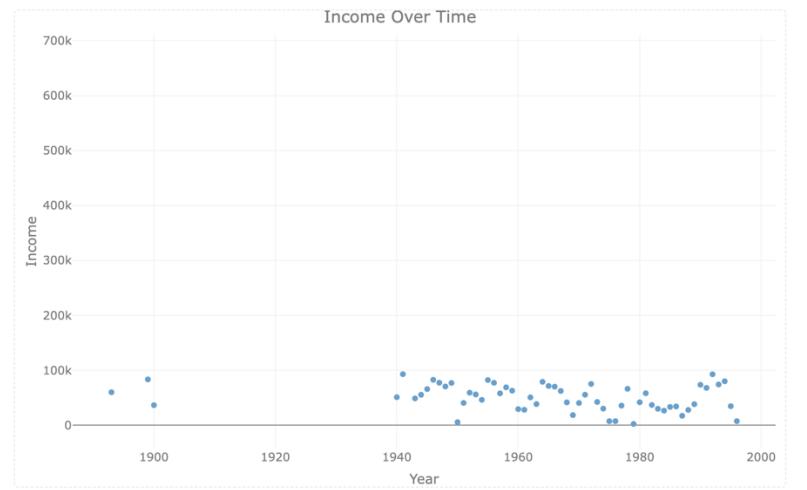


I created an interactive plot with an animating view

```
animated_plot <- marketing_campaign %>%
  plot_ly(x = ~Year_Birth, y = ~Income, type = "scatter", mode = "markers", ids =
  ~Year_Birth,
  animation_frame = ~Year_Birth, animation_group = ~Year_Birth)
```

```
animated_plot <- animated_plot %>%
  layout(title = "Income Over Time",
  xaxis = list(title = "Year"),
  yaxis = list(title = "Income"),
  showlegend = FALSE)
```

`animated_plot`



```
animating1 <- ggplot(marketing_campaign, aes(MntWines, Income, color =
Marital_Status)) + geom_point(aes(size = Complain, frame = Marital_Status)) +
scale_x_log10()
```

`animating1`

