scollini / Final_project

Watch 1 Star 0 Fork 0

branch: master Final_project / slave_network.Rmd

scollini 3 days ago Added R markdown file

1 contributor

253 lines (214 sloc) | 12.278 kb Raw Blame History

title	author	date	output
Network Analysis of the Enslaved Community at Mount Vernon, 1786-1799	Sara Collini	December 3, 2014	html_document

Introduction

This project uses network analysis in R to explore community amoung Mount Vernon slaves. The graph investigates two main communities: kinship relations and work relations. The information on the slaves draws from a compiled census created by the Mount Vernon staff at the Fred W. Smith National Library for the Study of George Washington.

First, we load the necessary libraries.

```
library(igraph)
library(dplyr)
library(tidyr)
library(stringr)
library(historydata)
library(ggplot2)
```

Data Manipulation

Read in the data

The data for this project exists in three csv files. The first file contains every slave with at least one kinship relation on Mount Vernon, from 1786-1799 .

```
slaverel <- read.csv("~//Desktop/Clio-3/finalproject/slaveRelationships.csv", stringsAsFactors = FALSE)
head(slaverel)
```

The second contains every slave and the respective static information, such as gender and owner, for 1786 and 1799.

```
total_slave <- read.csv("~//Desktop/Clio-3/finalproject/clean.csv", stringsAsFactors = FALSE) %>%
  select(id, Gender, Birth.Year, Owner)
head(total_slave)
```

This file contains both the 1786 and 1799 censuses with the relevant slaves. This means the slaves present in both

censuses are repeated but might have a different skill or farm.

```
slave_census <- read.csv("~//Desktop/Clio-3/finalproject/census.csv", stringsAsFactors = FALSE) %>%
  select(id, Gender, Birth.Year, Skill, Farm, Census, Owner)
head(slave_census)
```

Data Joining

Next, we must separate out the two most important and complete censuses: 1786 and 1799. We select only the variables that change between the two censuses.

```
slave_1786 <- slave_census %>%
  filter(Census == "1786") %>%
  select(id, Census, Skill, Farm)

slave_1799 <- slave_census %>%
  filter(Census == "1799") %>%
  select(id, Census, Skill, Farm)
```

Join the two censuses into one dataframe by the "id" variable in order to access that information on the network graph.

```
total_slave <- total_slave %>%
  left_join(slave_1786, by = "id") %>%
  left_join(slave_1799, by = "id")
head(total_slave)
```

Create a Graph Object

We create a graph object of the slaverel dataframe in order to graph the kinship relations between the slaves.

```
slave_graph <- graph.data.frame(slaverel, directed = "FALSE")
```

In order to add those slaves without kinship relations, we must add them as extra vertices to the graph object.

```
slave_graph <- slave_graph + vertex("Adam B", "Adam C", "Anthony A", "Austin A", "Bath A", "Bristol A", "Bruns
```

Now, we create another dataframe that lists all the vertex attributes for each slave in order to visualize them on the network graph.

```
slave_vertex <- data_frame(name = V(slave_graph)$name) %>%
  left_join(total_slave, by = c("name" = "id"))
head(slave_vertex)
```

Plotting the First Network Graph

```
igraph.options(vertex.size=3,
               edge.arrow.size=.5)
set.seed(33)
```

```
plot(slave_graph, vertex.label = NA)
```

Immediately, the overall kinship structure of the enslaved community is apparent from this network graph. Four family groups comprise one large familial relation and several smaller groups also exist. Many slaves had no kinship relations (no known kinship relations) and some slaves were only connected to one or two other people. In order to analyze this information further, we must be able to visualize the vertex attributes and the different kinship relationships.

Gender

In order to visualize the gender of the slaves, we can change the shape of each vertex accordingly.

```
V(slave_graph)$shape <- ifelse(slave_vertex$Gender == "Female", "circle",
                               ifelse(slave_vertex$Gender == "Male", "square",
                                      "rectangle"))
set.seed(33)
plot(slave_graph, vertex.label = NA)
```

Females are circles, while males are squares.

Relationships

In order to visualize the different kinships relations, we can colorize the edges according to type of relation with a function.

```
lookup_edge <- function(relationship) {
  if(relationship == "Spouse") return("red")
  if(relationship == "Child") return("orange")
  if(relationship == "Sibling") return("blue")
  return("gray")
}

lookup_edge <- Vectorize(lookup_edge, USE.NAMES = FALSE)

E(slave_graph)$color <- lookup_edge(E(slave_graph)$relationship)
set.seed(33)
plot(slave_graph,
     vertex.label = NA,
     vertex.size = 3)
```

The network graph is easier to decipher. We can see that the one large family is connected by a slave marriage. Many of the two person families are slave marriages, while others are mother and child. With this graph, we can count the number of each relationship and know where they exist within each family.

Evolution of Kinship Networks in Space

Since both slave censuses taken by Washington list the farms that each slave lived and worked on, we can use that variable to locate the slave families within the estate. We use a function similar to the edge coloring function:

```
lookup_color <- function(type) {
  if(is.na(type)) return("gray")
  if(type == "Mansion House") return("purple")
  if(type == "Muddy Hole") return("blue")
  if(type == "Dogue Run") return("green")
```

```
  if(type == "Union Farm") return("orange")
  if(type == "River Farm") return("yellow")
  if(type == "Mr. Lears") return("brown")
  if(type == "Mrs. Washington's") return("brown")
  if(type == "Cedar Grove") return("brown")
  return("gray")
}

lookup_color <- Vectorize(lookup_color, USE.NAMES = FALSE)
```

Five Farms made up the Mount Vernon estate: Mansion House, Muddy Hole, Dogue Run, Union Farm, and River Farm. Most slaves married and had kinship relations within Washington's slave community. However, some slaves married people owned by others. I have colored those non-Mount Vernon slave spouses brown. The gray color represents slaves that were not active on each census, meaning that they were not born yet or had died.

Slave Families Across Farms in 1786

```
V(slave_graph)$color <- lookup_color(slave_vertex$Farm.x)
set.seed(33)

plot(slave_graph,
     vertex.label = NA,
     vertex.size = 3)
title("Slave Families on Mount Vernon Farms in 1786")
legend("bottomleft", legend = c("Mansion House", "Muddy Hole", "Dogue Run", "Union Farm", "River Farm", "Other
       col = c("purple", "blue", "green", "orange", "yellow", "brown", "gray"), pch = 19,
              title = "Farm")
```

This graph tells us a lot more information. We can place the slave families in space on the estate and organize the various family structures.

1786 Family Structures

From this graph, four types of family structures arise: singles, one parent, two parent, and couples with no known children. We can plot the numbers of each family structure for the entire estate, as well as the individual farms. We will use ggplot2 bar graphs to plot this information drawn from the network graph into a csv.

```
structures <- read.csv("~//Desktop/Clio-3/finalproject/FamilyStructures.csv", stringsAsFactors = FALSE)
head(structures)
```

We can use a simple bar plot to show the percentages of each family structure:

```
structures_1786 <- structures %>%
  filter(Census == "1786")

ggplot(data = structures_1786, aes(x = Family.structure, y = Percent)) + facet_wrap(~Farm) + geom_bar(stat = '
```

From this plot, we can see that almost every farm and the entire estate had a majority of single person structures.

Slave Families Across Farms in 1799

```
V(slave_graph)$color <- lookup_color(slave_vertex$Farm.y)
set.seed(33)
```

```
plot(slave_graph,
     vertex.label = NA,
     vertex.size = 3)
title("Slave Families on Mount Vernon Farms in 1799")
legend("bottomleft", legend = c("Mansion House", "Muddy Hole", "Dogue Run", "Union Farm", "River Farm", "Other
       col = c("purple", "blue", "green", "orange", "yellow", "brown", "gray"), pch = 19,
       title = "Farm")
```

1799 Family Structures

Let's plot the four family structures for 1799 next to the 1786 figures:

```
ggplot(data = structures, aes(x = Family.structure, y = Percent, fill = Census)) + facet_wrap(~Farm) + geom_ba
```

Immediately, we see the increase in the percentage of two-parent family strucutres in just 13 years. Single person structures still dominate on Mansion House and Union Farm, but every farm and the overall estate increases in two-parent families.

Community Detection

We can use the community detection methods in R in order to focus on each family grouping in 1799.

```
comm <- walktrap.community(slave_graph)
length(comm)
```

The walktrap.community DESCRIBE and calculates 119 slave family groups on Mount Vernon in 1799. This is not completely accurate, as the largest family is counted as 4 groups. A more nuanced analysis of the number and definition of family groupings is needed for this data, but the exploring the individual groups is helpful.

We can also calculate the size of each of the family groups.

```
sizes(comm)
```

We can inspect each community in the graph and list the names of each family member.

```
memb <- membership(comm)
names(memb[memb ==2])
```

We can also show the kinship network graph colored by each family group.

```
set.seed(33)
plot(slave_graph, vertex.label = NA, vertex.color = memb, vertex.size = 3)
title("Families detected by walktrap.community()")
```

```
set.seed(33)
plot(comm, slave_graph, vertex.label = NA, vertex.size = 2)
title("Families detected by walktrap.community()")
```

Family Tree

In order to see how each slave is connected within a family, we can show a subgraph of the larger network graph in a different layout. We will use community 2 for this example.

```
induced.subgraph(slave_graph, which(membership(comm) ==2)) %>%
  plot(layout = layout.reingold.tilford, vertex.label = names(memb[memb == 2]))
title("Community 2 among Mount Vernon")
legend("bottomleft", legend = c("Spouse", "Child", "Sibling"),
       col = c("red", "orange", "blue"), pch = 19,
       title = "Relationship")
```

This graph is not the most visually appealing, nor does it show a true family tree in chronological and hierarchical order. Suckey Bay and NA9 are married and have three children: Nancy G, Nancy H, and Rose A. Rose A has four childern of her own: Hagar A, Joe D, Simon A, and Tom C. This entire family does reside on River Farm in 1799, except Suckey Bay's spouse, who is unknown. This subgraph does show a two generational family living and working on the same farm in 1799. They are all connected by their familial ties, as well as their working community.

Let's explore a smaller family

```
induced.subgraph(slave_graph, which(membership(comm) == 32)) %>%
  plot(layout = layout.reingold.tilford, vertex.label = names(memb[memb == 32]))
title("Community 32 among Mount Vernon")
legend("bottomleft", legend = c("Spouse", "Child", "Sibling"),
       col = c("red", "orange", "blue"), pch = 19,
       title = "Relationship")
```

This graph shows that Daphne B and Simms A are spouses with three children: Maria A, Matilda B, and Polly A. The women live on Union Farm, but Simms A lives on Mansion House farm. This means that Daphne B and Simms A would have traveled across these two farms in order to see each other. Simms A belongs to two different communities: his family and the Mansion House working community.