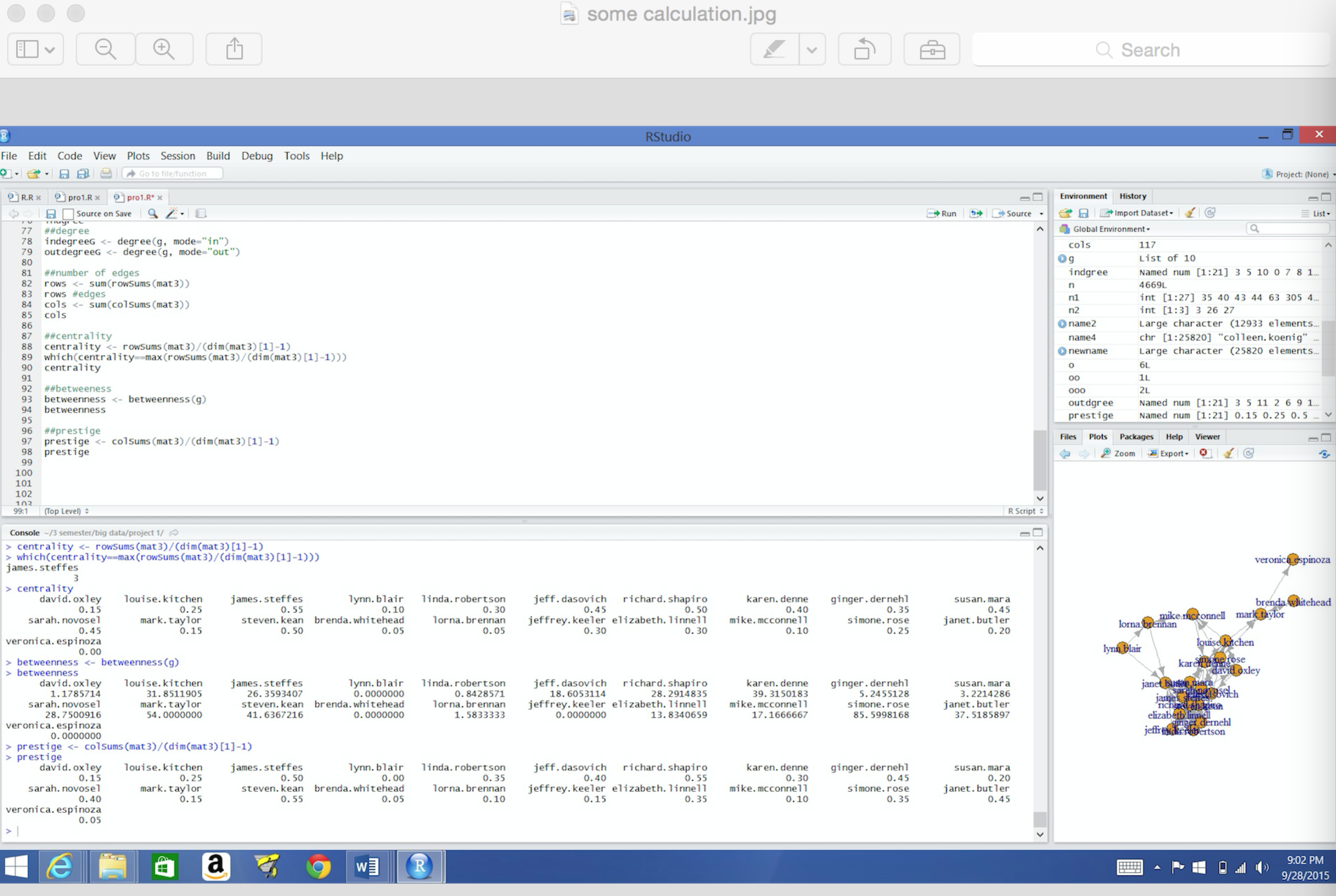
## Centrality, Betweeness, and Prestige



|  |  |  |  |
| --- | --- | --- | --- |
|  | centrality | betweenness | prestige |
| david.oxley | 0.15 | 1.178571429 | 0.15 |
| louise.kitchen | 0.25 | 31.85119048 | 0.25 |
| james.steffes | 0.55 | 26.35934066 | 0.5 |
| lynn.blair | 0.1 | 0 | 0 |
| linda.robertson | 0.3 | 0.842857143 | 0.35 |
| jeff.dasovich | 0.45 | 18.60531136 | 0.4 |
| richard.shapiro | 0.5 | 28.29148352 | 0.55 |
| karen.denne | 0.4 | 39.31501832 | 0.3 |
| ginger.dernehl | 0.35 | 5.245512821 | 0.45 |
| susan.mara | 0.45 | 3.221428571 | 0.2 |
| sarah.novosel | 0.45 | 28.75009158 | 0.4 |
| mark.taylor | 0.15 | 54 | 0.15 |
| steven.kean | 0.5 | 41.63672161 | 0.55 |
| brenda.whitehead | 0.05 | 0 | 0.05 |
| lorna.brennan | 0.05 | 1.583333333 | 0.1 |
| jeffrey.keeler | 0.3 | 0 | 0.15 |
| elizabeth.linnell | 0.3 | 13.83406593 | 0.35 |
| mike.mcconnell | 0.1 | 17.16666667 | 0.1 |
| simone.rose | 0.25 | 85.59981685 | 0.35 |
| janet.butler | 0.2 | 37.51858974 | 0.45 |
| veronica.espinoza | 0 | 0 | 0.05 |

* Red denotes the people with the highest value in that category
  + James.Steffes has the highest centrality (central person in graph)
  + Simone.Rose has the highest betweenness
  + Richard.Shapio and Steven.kean are tied for the highest prestige

## Graph of Adjacency Matrix

## 

## Adjacency Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | david.oxley | louise.kitchen | james.steffes | lynn.blair | linda.robertson | jeff.dasovich | richard.shapiro |
| david.oxley | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| louise.kitchen | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| james.steffes | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| lynn.blair | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| linda.robertson | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| jeff.dasovich | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| richard.shapiro | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| karen.denne | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| ginger.dernehl | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| susan.mara | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| sarah.novosel | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| mark.taylor | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| steven.kean | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| brenda.whitehead | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| lorna.brennan | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jeffrey.keeler | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| elizabeth.linnell | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| mike.mcconnell | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| simone.rose | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| janet.butler | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| veronica.espinoza | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Adjacency Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | karen.denne | ginger.dernehl | susan.mara | sarah.novosel | mark.taylor | steven.kean | brenda.whitehead |
| david.oxley | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| louise.kitchen | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| james.steffes | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| lynn.blair | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| linda.robertson | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| jeff.dasovich | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| richard.shapiro | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| karen.denne | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| ginger.dernehl | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| susan.mara | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| sarah.novosel | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| mark.taylor | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| steven.kean | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| brenda.whitehead | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| lorna.brennan | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| jeffrey.keeler | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| elizabeth.linnell | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| mike.mcconnell | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| simone.rose | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| janet.butler | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| veronica.espinoza | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Adjacency Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | lorna.brennan | jeffrey.keeler | elizabeth.linnell | mike.mcconnell | simone.rose | janet.butler | veronica.espinoza |
| david.oxley | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| louise.kitchen | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| james.steffes | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| lynn.blair | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| linda.robertson | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| jeff.dasovich | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| richard.shapiro | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| karen.denne | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| ginger.dernehl | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| susan.mara | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| sarah.novosel | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| mark.taylor | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| steven.kean | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| brenda.whitehead | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| lorna.brennan | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| jeffrey.keeler | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| elizabeth.linnell | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| mike.mcconnell | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| simone.rose | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| janet.butler | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| veronica.espinoza | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**degree**

|  |  |  |
| --- | --- | --- |
|  | indgree | outdgree |
| david.oxley | 3 | 3 |
| louise.kitchen | 5 | 5 |
| james.steffes | 10 | 11 |
| lynn.blair | 0 | 2 |
| linda.robertson | 7 | 6 |
| jeff.dasovich | 8 | 9 |
| richard.shapiro | 11 | 10 |
| karen.denne | 6 | 8 |
| ginger.dernehl | 9 | 7 |
| susan.mara | 4 | 9 |
| sarah.novosel | 8 | 9 |
| mark.taylor | 3 | 3 |
| steven.kean | 11 | 10 |
| brenda.whitehead | 1 | 1 |
| lorna.brennan | 2 | 1 |
| jeffrey.keeler | 3 | 6 |
| elizabeth.linnell | 7 | 6 |
| mike.mcconnell | 2 | 2 |
| simone.rose | 7 | 5 |
| janet.butler | 9 | 4 |
| veronica.espinoza | 1 | 0 |

* Red denotes the people with the highest value in that category

**Maximum Path**

|  |  |  |  |
| --- | --- | --- | --- |
|  | david.oxley | lynn.blair | sarah.novosel |
| david.oxley | 0 | 8 | 1 |
| louise.kitchen | 1 | 7 | 2 |
| james.steffes | 3 | 3 | 4 |
| lynn.blair | 0 | 0 | 0 |
| linda.robertson | 4 | 4 | 5 |
| jeff.dasovich | 6 | 6 | 7 |
| richard.shapiro | 5 | 5 | 6 |
| karen.denne | 2 | 10 | 3 |
| ginger.dernehl | 7 | 12 | 8 |
| susan.mara | 8 | 11 | 9 |
| sarah.novosel | 9 | 13 | 0 |
| mark.taylor | 14 | 10 | 14 |
| steven.kean | 10 | 14 | 10 |
| brenda.whitehead | 15 | 11 | 15 |
| lorna.brennan | 15 | 1 | 15 |
| jeffrey.keeler | 11 | 15 | 11 |
| elizabeth.linnell | 12 | 16 | 12 |
| mike.mcconnell | 14 | 10 | 14 |
| simone.rose | 13 | 9 | 13 |
| janet.butler | 16 | 2 | 16 |
| veronica.espinoza | 15 | 11 | 15 |

* Red represent the maximum path length.

David. Oxley, Jynn Blair, Sarah Novosal are able to get the maximum path length.

* The first maximum path: David Oxley, Louise Kitchen, Karen Denne, James Steffes, Linda Robertson, Richard Shapiro, Jeff Dasovich, Ginger Dernehl, Susan Mara, Sarah Novosel, Steven Kean, Jeffrey Keller, Elizabeth Linnell, Simone Rose, Mike Mcconnell, Lorna Brennan, Janet Butler
* The second maximum path: Lynn Blair, Lorna Brennan, Janet Butler, James Steffes, Linda Robertson, Richard Shapiro, Jeff Dasovich, Louise Kitchen, David Oxley, Simone Rose, Karen Denne, Susan Mara, Ginger Dernehl, Sarah Novosel, Steven Kean, Jeffrey Keeler, Elizabeth Linnell.
* The third maximum path: Sarah Novosel, David Oxley, Louise Kitchen, Karen Denne, James Steffes, Linda Robertson, Richard Shapiro, Jeff Dasovich, Ginger Dernehl, Susan Mara, Steven Kean, Jeffrey Keeler, Elizabeth Linnell, Simone Rose, Mike Mcconnell, Lorna Brennan, Janet Butler.

# R Libraries used:

|  |  |
| --- | --- |
| Gsubfn | R package used for string and pattern matching |
| Proto | Object-oriented programming model. |
| Sqldf | Allows used to perform SQL selects on R data frames |
| igraph | Provides graphing functions |

# functions created:

|  |  |
| --- | --- |
| readEnron | Recursively reads all of the emails starting a the top level directory, extracts the To, From, Message-ID, Subject, and creates CVS files for each tag |
| buildMatrix | Unused |

# R functions used:

|  |  |
| --- | --- |
| degree | Determines the number of in and out degrees |
| betweenness | Computes the number of shortest paths going through a node |
| graph.adjacency | Create a graph from the adjacency matrix |
| matrix | Create a matrix from other data types |
| gsub | gsub perform replacement when a match is found |
| sqldf | Read a file into R filtering using an sql statement. |
| read.table | Reads a file in table |
| dfs | Calculate the maximum path length |