Minnesota Migration Network Analysis

Scott Chase

March 2016

1 Note

This paper is a final write-up for an Undergraduate Research Opportunities Project (UROP) corresponding to an analysis of Kickstarter network data. However, the data I worked with were both confidential and the subject of future publications by those who had collected and supervised the data. As such, I was unable to publish anything directly about this data. Instead, as I have agreed to with the UROP Office at the University of Minnesota, I will submit this report outlining a similar analysis of migration data, but treated as a social network. On previous projects, I have worked to gather and organize this data into this specific format (called an Adjacency Matrix). I should also note that there was a significant data cleaning task required for the Kickstarter data. I completed the data cleaning for the Minnesota Migration data in a previous UROP project. As such, only the migration network analysis will be shown in this document.

2 Introduction

2.1 Data Description

In a previous project, I had compiled county level migration data for the entire United States for the years 1992-2011. The result of data made public by the IRS based off Tax Records, I have compiled a large matrix that shows the number of people moving between any two given counties in the U.S. It shows both directions on movement (i.e., Hennepin County to Ramsey County, and Ramsey County to Hennepin County), as well as the number of people that moved but stayed within the same county during that year. As there are over 3000 counties in the United States, this results in a 9-million element matrix ($\approx 3000^2$). This is a very large network to analyze, so as a result, I will simply look at the migration network within Minnesota to simplify the analysis.

2.2 Data Example

Below is an example of the 2010-2011 tax season migration data for Minnesota.

```
mn_1011[1:10,1:10]

##       27001  27003 27005 27007 27009 27011 27013 27015 27017 27019
## 27001  5582     17    NA    NA    NA    NA    NA    NA    NA    NA
## 27003    29 123487    NA    20    20    NA    33    NA    18    32
## 27005    NA     NA 11827    17    NA    NA    NA    NA    NA    NA
## 27007    NA     12    18 14520    NA    NA    NA    NA    NA    NA
## 27009    NA     23    NA    NA 13796    NA    NA    NA    NA    NA
## 27011    NA     NA    NA    NA    NA  1958    NA    NA    NA    NA
```

```
## 27013    NA     23    NA    NA    NA    NA 21684    49    NA    24
## 27015    NA     NA    NA    NA    NA    NA    48 10422    NA    NA
## 27017    NA     10    NA    NA    NA    NA    NA    NA 12463    NA
## 27019    NA     24    NA    NA    NA    NA    18    NA    NA 31332
```

The first two digits of the row/column names are the state designator. For Minnesota, it is "27." The next three digits indicate the county. Together, these compromise a FIPS Code - used to identify geographic areas. County "27001" is Aitkin County, MN. County "27003" is Anoka County. A helpful mnemonic to read these matrices is "from row to column." Looking in the top left of the matrix, we see that 29 people moved from Anoka County to Aitkin County, 17 people moved from Aitkin County to Anoka County, 5,582 people moved around *within* Aitkin County, and 123,487 people moved around *within* Anoka County. For privacy concerns, the IRS does not report a value in any of these cells if under 10 people moved between those two counties. This is why there are so many NA values in the matrix. There isn't really anything we can do about that, so I will just replace those with 0.

```
mn_1011[is.na(mn_1011)]<-0
```

3 Analysis

3.1 Some Useful Definitions

- **Vertex/Node** An element in the network. In this case, each vertex/node is an entire county. A vertex/node is the unit in a network. For instance, in other contexts it could be birds, people, or even academic papers!

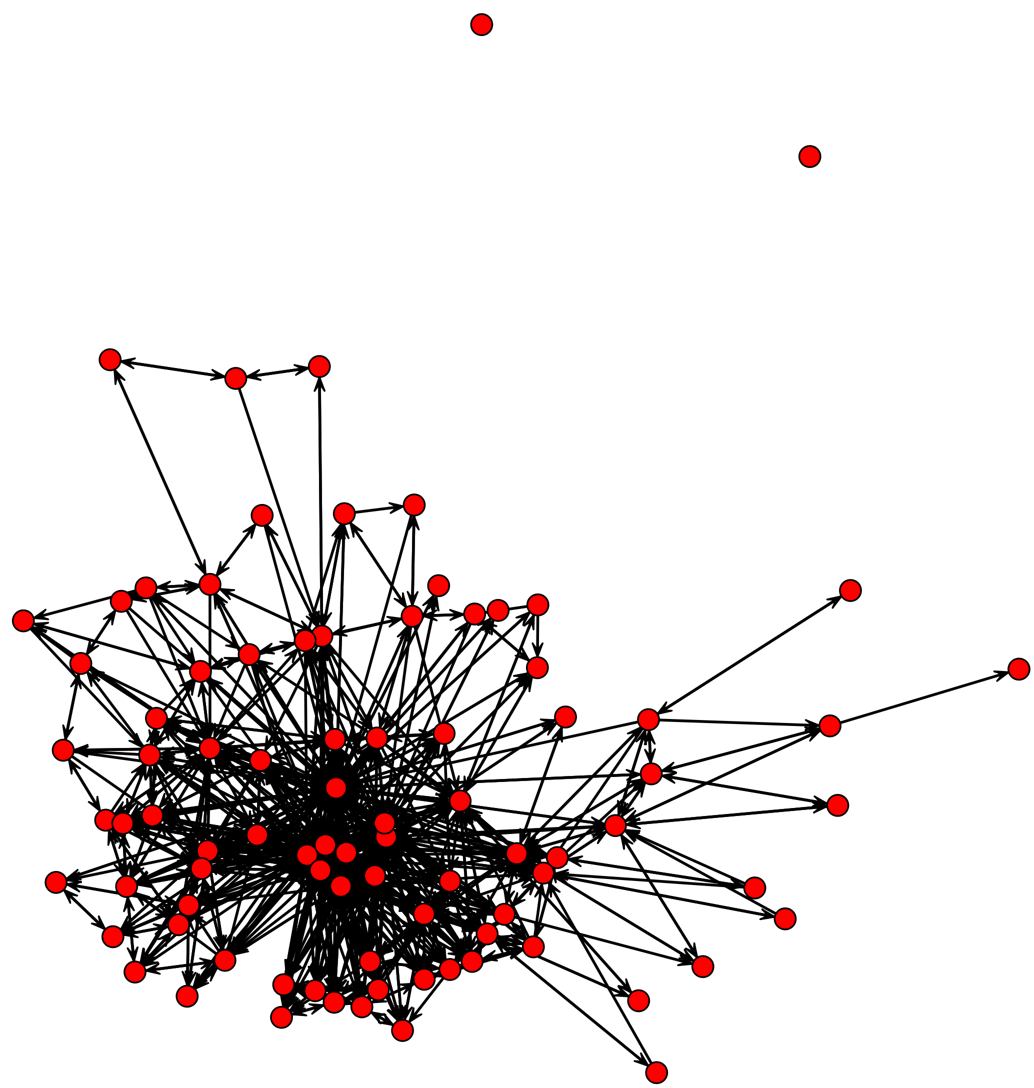
- **Edge** A link between two vertices/nodes. In the plots you will see below, the edges are represented by the lines between nodes. These can be directional, or non-directional, depending on whether or not the network is directional. A non-directional network just means that if node A has an edge with node B, then, automatically node B has an edge with node A. In this case, our network is directional. If people move from county A to county B, that does not necessarily imply that anyone moved from county B to county A.

- **Degree** Each vertex/node in the network will have a degree number. This is the number of edges between it and other vertices/nodes in the network.

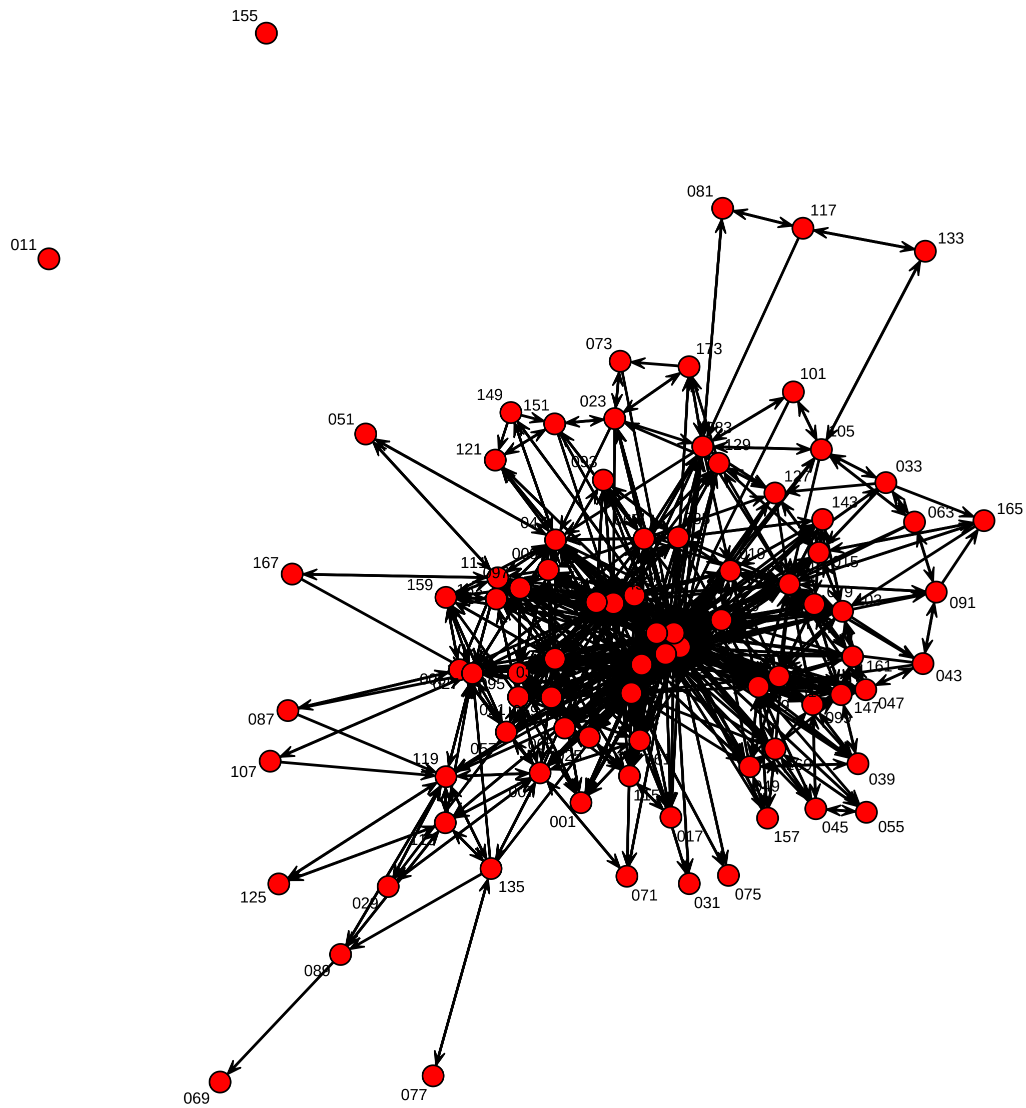
3.2 2010/2011 Minnesota Migration Network Example

First, let's create a simple network plot using the *statnet* package.

```
par(mar=rep(0,4))
set.seed(1)
v1=substr(colnames(mn_1011), 3, 5)
mn_net_1011<-network(mn_1011, directed=T)
plot(mn_net_1011, arrowhead.cex=0.7)
```

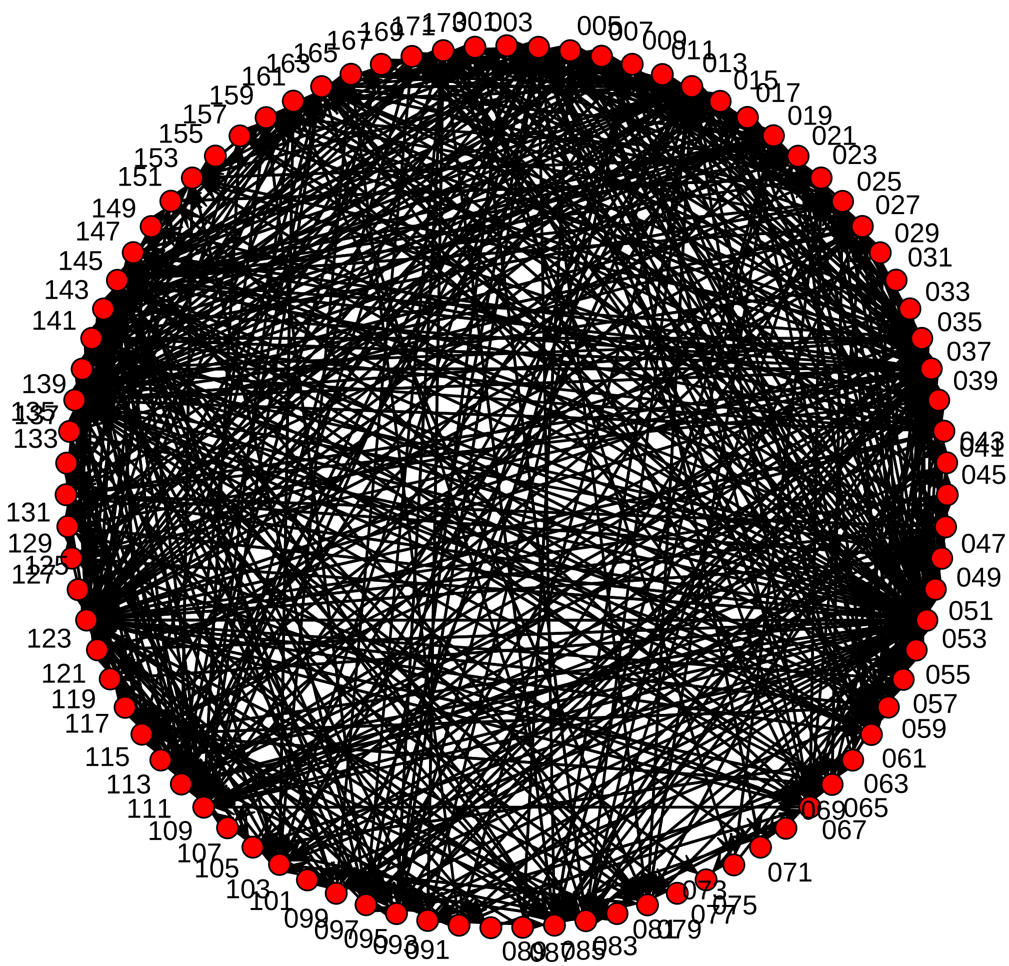


```r
plot(mn_net_1011, label=v1, arrowhead.cex=0.7, label.cex=0.6)
```



The reason there are two plots is that due to our network having 87 vertices, and 965 total edges, it is tricky to visualize. One includes the FIPS Codes and the other does not. From the graph, you can see that just about every county has a link between itself and another county - except for two, namely county “27011” and county “27155.” These correspond to Big Stone County, and traverse county, respectively. Maybe we can simplify this a little bit. It is often recommended that for networks with many nodes (such as this one) a circle layout be used. This method can be useful when there are many connections, as it usually isn't as clustered.

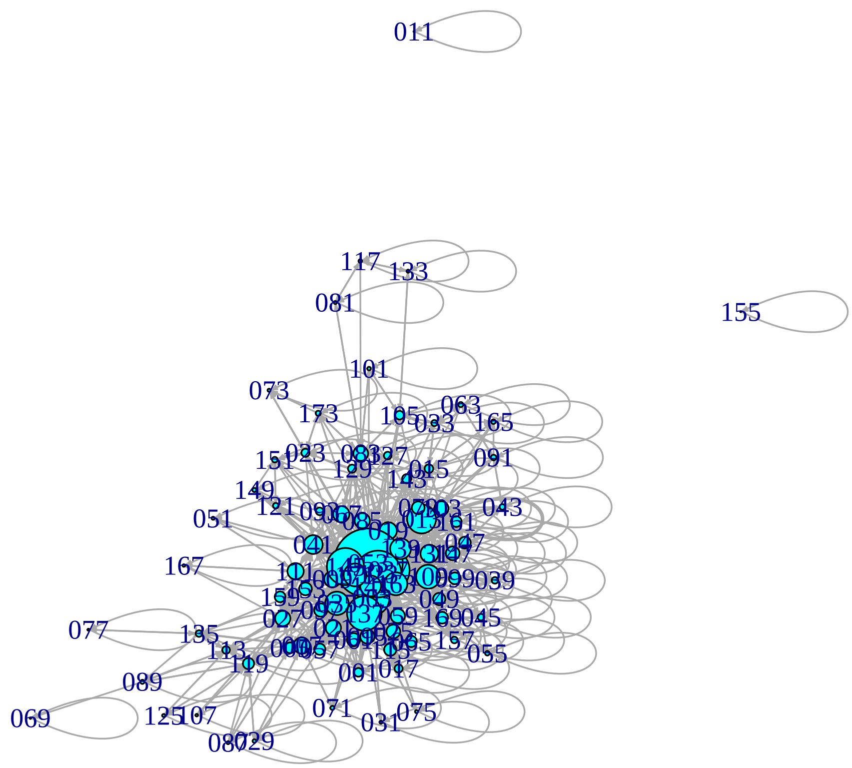
```r
par(mar=rep(0,4))
plot.network(mn_net_1011, mode="circle", label=v1)
```



That really isn't much better. Below is a similar network plot, but with two differences. First, I have set the size of the vertices to be proportional to the degree of each node in the network, but divided by 10 as just using the raw degree as a size made some of the nodes much too large and they covered up other nodes. Secondly, I have used the fruchterman reingold node placement method. This is just a method for placing the nodes on the plot.

Another Note: I will use a function written by Michael Hahsler called `map`. What this does is take a vector, and create a map to a smaller range. This is for scaling purposes. If I made the nodes or edges too large on the graph, it would be hard to visualize. I also made use of Michael Hahsler's `igraph` package examples, as he has a very good set of examples of the package (Hahsler, 2014).

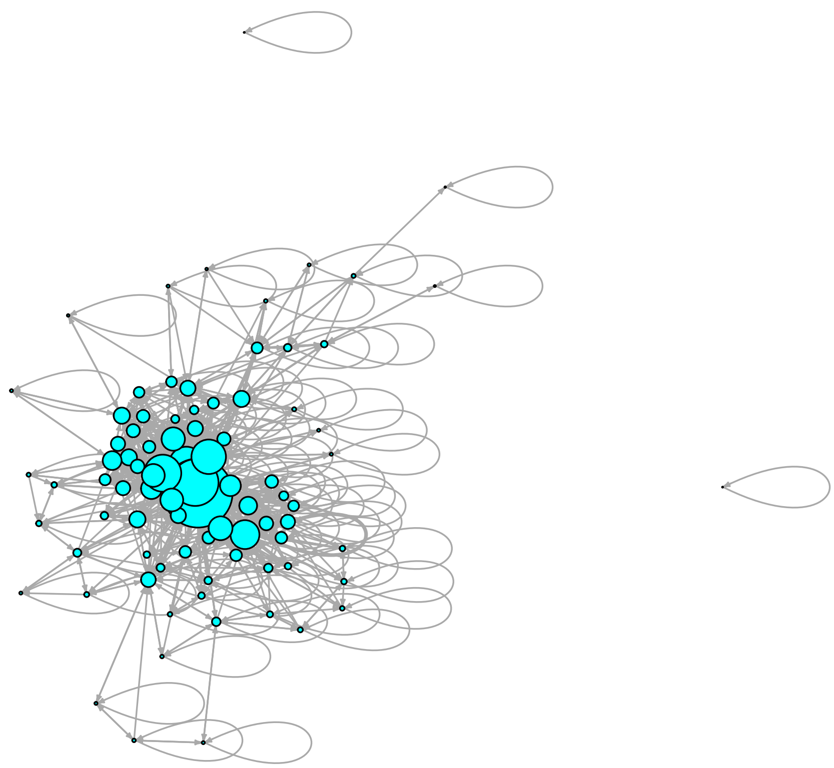
```
source("http://michael.hahsler.net/SMU/ScientificCompR/code/map.R")
g_1011<-graph.adjacency(mn_1011, mode="directed", weighted=T,
                        diag=T, add.colnames = T)
plot(g_1011, layout=layout.fruchterman.reingold(g_1011),
      vertex.size=igraph::degree(g_1011)/7,
      vertex.label=v1, edge.arrow.size=.2,
      vertex.color="cyan",edge.width=map(c(mn_1011), c(1,5)))
```



It is a little hard to see which counties are which, given the clustering of labels. Below is the same plot without labels. I am using the `map` function to rescale the relative sizes of the migrations. The $c(mn\_1011)$ command simply takes the adjacency matrix and turns it into a vector, and maps it to the set of real numbers $[1, 5]$. This plot now takes into account the relative *sizes* of the migrations.

```
plot(g_1011, layout=layout.fruchterman.reingold(g_1011),
  vertex.size=igraph::degree(g_1011)/7, vertex.label=NA,
  edge.arrow.size=.2, vertex.color="cyan",
```

```
  edge.width=map(c(mn_1011), c(1,5)))
```



Note that there are a lot of counties in the middle of the network, centered around the largest nodes in the network. These are the counties with the highest degrees in the network, as shown by the sizes of the nodes. It is no accident that these are the nodes in the center of the network. Around the fringes of the center of the network, there are some more counties with smaller degrees. Almost every county is appearing in one, large cluster.

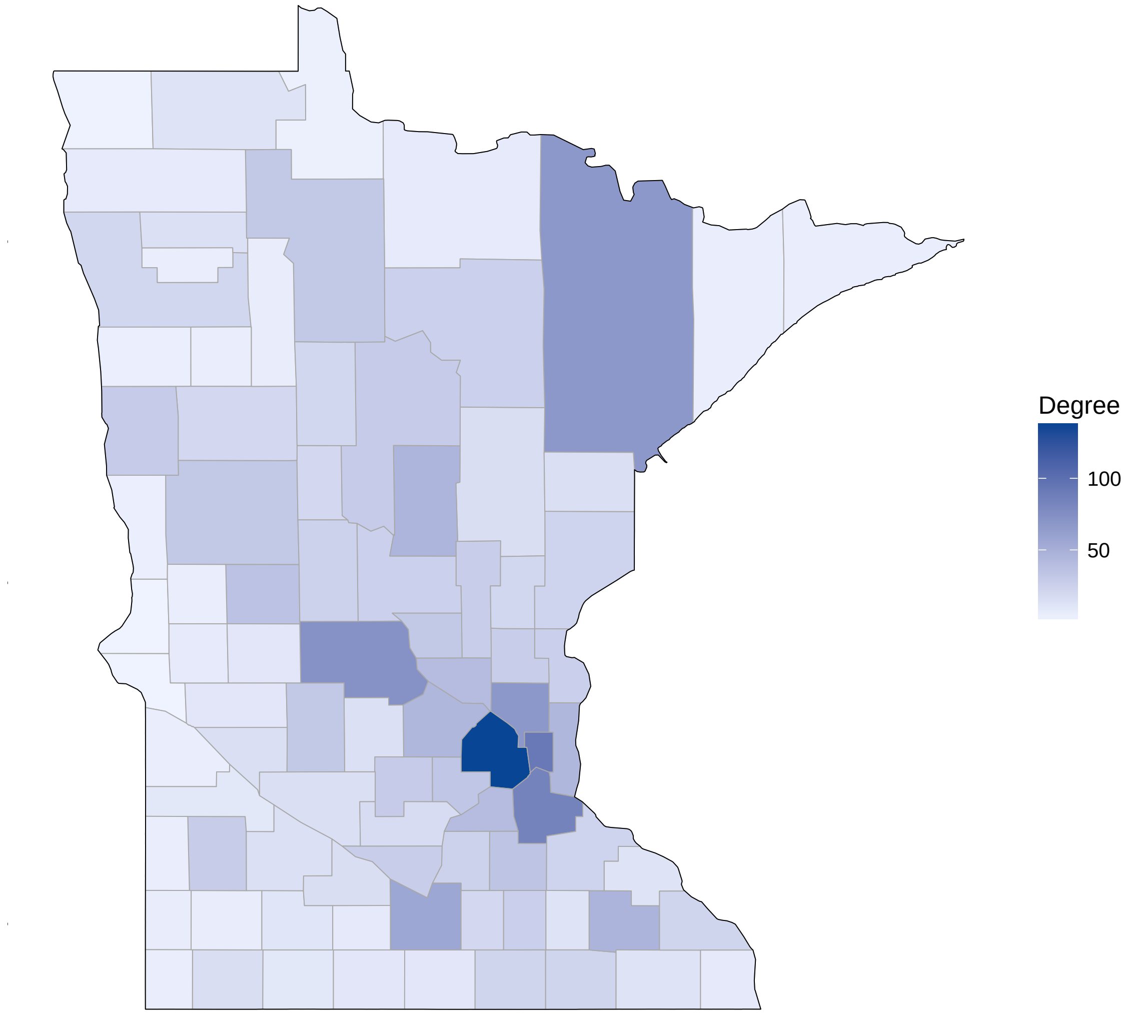
```
(d<-igraph::degree(g_1011))

##  [1]  17  68  21  32  32   2  57  17  16  34  30  16  27  30   7   6  12
## [18]  46  84  13  37  11  13  23  24   6 136   9  22  28  26  10  22  32
## [35]   3   8   6   6   4  25   6  29  30   6   8  11  15  28  26  23   7
## [52]  28  17   5  47  32  15  24   7  22  11  92   6  15  16  35   6  13
## [69]  68  41  41  18  73  27   8  11  25   2  13  21  21  45   9   5  23
## [86]  45  10

col1=cbind(paste("27", v1, sep="")[1:30], d[1:30])
col2=cbind(paste("27", v1, sep="")[31:60], d[31:60])
col3=cbind(c(paste("27", v1, sep="")[61:87], rep(NA, 3)),
        c(d[61:87], rep(NA, 3)))
x=data.frame(col1, col2, col3)
colnames(x)=c("FIPS1", "Degree1", "FIPS2", "Degree2", "FIPS3", "Degree3")
mncounty<-countyfips2010[substr(countyfips2010$fips,1,2)=="27",]
x$FIPS1=mncounty$countyname[match(x$FIPS1,
          countyfips2010[substr(countyfips2010$fips, 1, 2)=="27",1])]
x$FIPS2=mncounty$countyname[match(x$FIPS2,
          countyfips2010[substr(countyfips2010$fips, 1, 2)=="27",1])]
x$FIPS3=mncounty$countyname[match(x$FIPS3,
            countyfips2010[substr(countyfips2010$fips, 1, 2)=="27",1])]
x

##                FIPS1 Degree1                  FIPS2 Degree2
## 1      aitkin county      17          itasca county      26
## 2       anoka county      68         jackson county      10
## 3      becker county      21         kanabec county      22
## 4    beltrami county      32       kandiyohi county      32
## 5      benton county      32         kittson county       3
## 6   big stone county       2     koochiching county       8
## 7  blue earth county      57   lac qui parle county       6
## 8       brown county      17            lake county       6
## 9     carlton county      16 lake of the woods county       4
## 10     carver county      34        le sueur county      25
## 11       cass county      30         lincoln county       6
## 12   chippewa county      16            lyon county      29
## 13    chisago county      27          mcleod county      30
## 14       clay county      30        mahnomen county       6
## 15 clearwater county       7        marshall county       8
## 16       cook county       6          martin county      11
```

```
## 17 cottonwood county      12          meeker county      15
## 18  crow wing county      46      mille lacs county      28
## 19     dakota county      84        morrison county      26
## 20      dodge county      13           mower county      23
## 21    douglas county      37          murray county       7
## 22  faribault county      11        nicollet county      28
## 23   fillmore county      13          nobles county      17
## 24   freeborn county      23          norman county       5
## 25    goodhue county      24         olmsted county      47
## 26      grant county       6      otter tail county      32
## 27   hennepin county     136      pennington county      15
## 28    houston county       9            pine county      24
## 29    hubbard county      22       pipestone county       7
## 30     isanti county      28            polk county      22
##                       FIPS3 Degree3
## 1            pope county      11
## 2          ramsey county      92
## 3        red lake county       6
## 4         redwood county      15
## 5        renville county      16
## 6            rice county      35
## 7            rock county       6
## 8          roseau county      13
## 9       st. louis county      68
## 10          scott county      41
## 11      sherburne county      41
## 12         sibley county      18
## 13        stearns county      73
## 14         steele county      27
## 15        stevens county       8
## 16          swift county      11
## 17           todd county      25
## 18       traverse county       2
## 19        wabasha county      13
## 20         wadena county      21
## 21         waseca county      21
## 22     washington county      45
## 23       watonwan county       9
## 24         wilkin county       5
## 25         winona county      23
```

```
## 26          wright county      45
## 27 yellow medicine county      10
## 28                   <NA>    <NA>
## 29                   <NA>    <NA>
## 30                   <NA>    <NA>
```

Perhaps not surprisingly, the two counties in Minnesota with the highest degrees are Hennepin and Ramsey County, respectively. For those unfamiliar with Minnesota, these are the two counties that comprise the largest metropolitan area in the state. Next, I will look at a map of the state, by degree.

```
df=rbind(col1, col2, col3)
#remove the NAs
df=df[1:87,]
colnames(df)<-c("region", "value")
df<-as.data.frame(df, stringsAsFactors=F)
df[,1]<-as.numeric(df[,1])
df[,2]<-as.numeric(df[,2])
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```



Although this map does not show the magnitude or direction of the migration patterns between counties, it does show that most of the counties with the highest degrees are around the metro area. The one exception is the large county in the northeast of the state, namely St. Louis County, with 68 connections. This is likely because Duluth, another fairly large city, is located in this county. Interestingly, it doesn't seem that geographic proximity has much to do with state-wide migration. Counties all over the entire state have non-zero degrees, and from the network plot above, only two counties were not connected to other counties. While the counties with the most nodes are mostly geographically close to each other, they all have much larger degrees, and the degree does not take into account the size of the edge, only the number of edges. In essence, what we are seeing is that many people are moving to and from the largest counties from all areas of the state, regardless of size, geography, or proximity.

4 Migration Networks Across Time

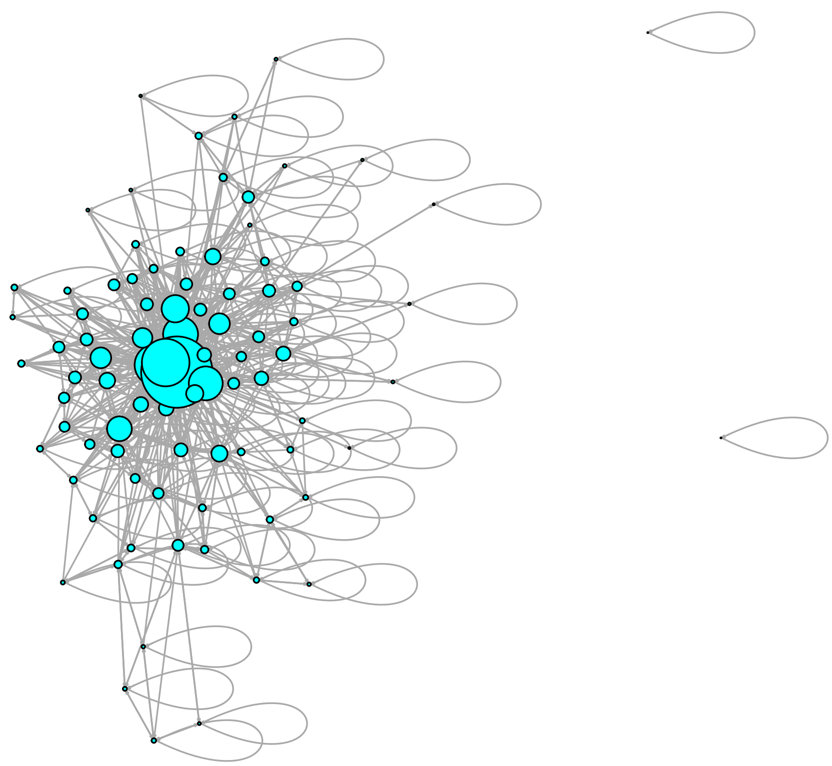
As part of the previous UROP projects I've worked on, I have access to this same set of data, but for different tax seasons, spanning the years 1992 to 2011. Now that I've settled on a decent method for visualizing these networks, I will compare them across time and see how the networks change. As the code for loading and generating all of these networks is very, very lengthy and exactly the same as before, I will not show it in this document. At this point, loaded into my `R` workspace are graph network objects, with names such as g_0405, to designate the tax season for which the network was established.

4.1 Note

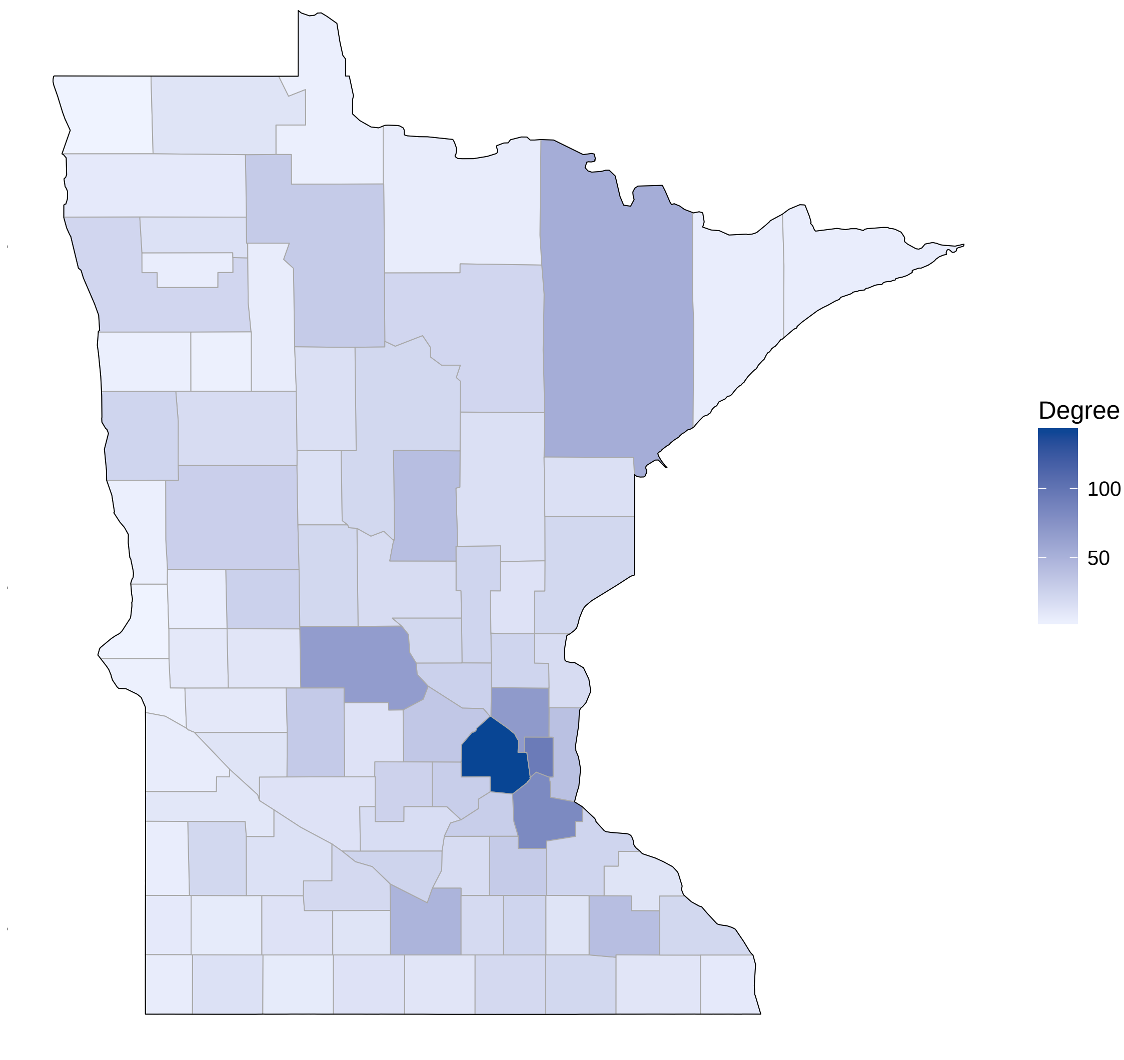
I have elected to not include the FIPS codes as labels on the network graph. There are two reasons for this. First, labeling with the FIPS codes makes the graphs unnecessarily crowded and difficult to view. Secondly, adding the FIPS code, which is only an arbitrary numeric identifier for the county, does not give the viewer any sort of information as to which county corresponds to the node in the network. To help the viewer visualize the network and the degree of each node, I will include choropleth maps of Minnesota for each tax season, shaded by the degree of the node corresponding to that county. A darker colored county corresponds to a larger node in the network (as measured by how many edges it has, i.e. the degree). It is important to note that while the choropleth map does take into account the number of connections to other counties, it does not take into account the sizes of those connections. The thickness of the lines on the network graph measure the magnitude of these connections between any two given counties.

4.2 1992-1993

```
plot(g_9293, layout=layout.fruchterman.reingold(g_9293),
  vertex.size=igraph::degree(g_9293)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_9293), c(1,5)))
```

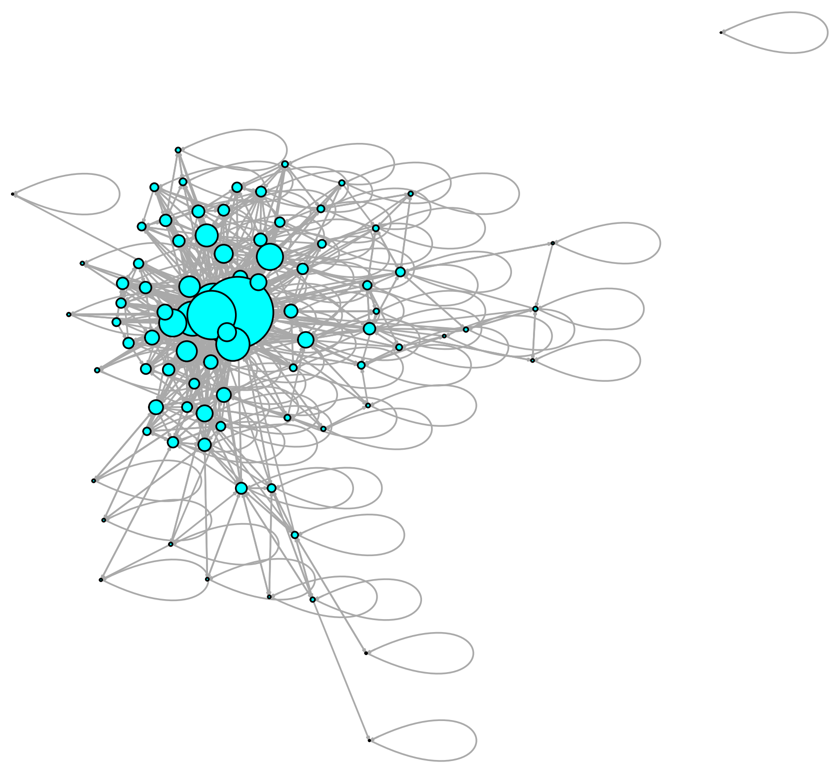


```
v2=paste("27", v1, sep="")
d_9293<-igraph::degree(g_9293)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_9293))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

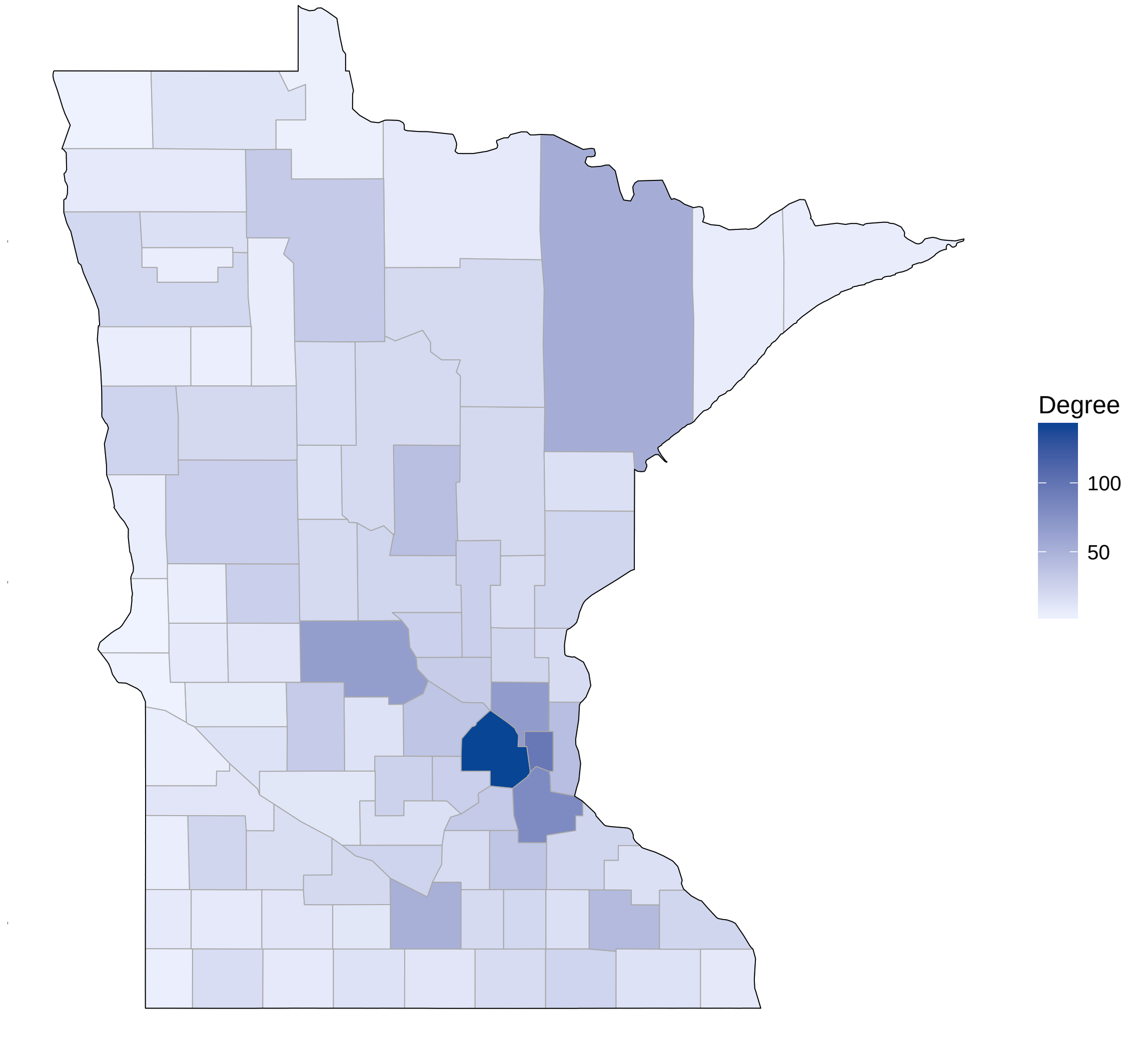


4.3 1993-1994

```
plot(g_9394, layout=layout.fruchterman.reingold(g_9394),
  vertex.size=igraph::degree(g_9394)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_9394), c(1,5)))
```

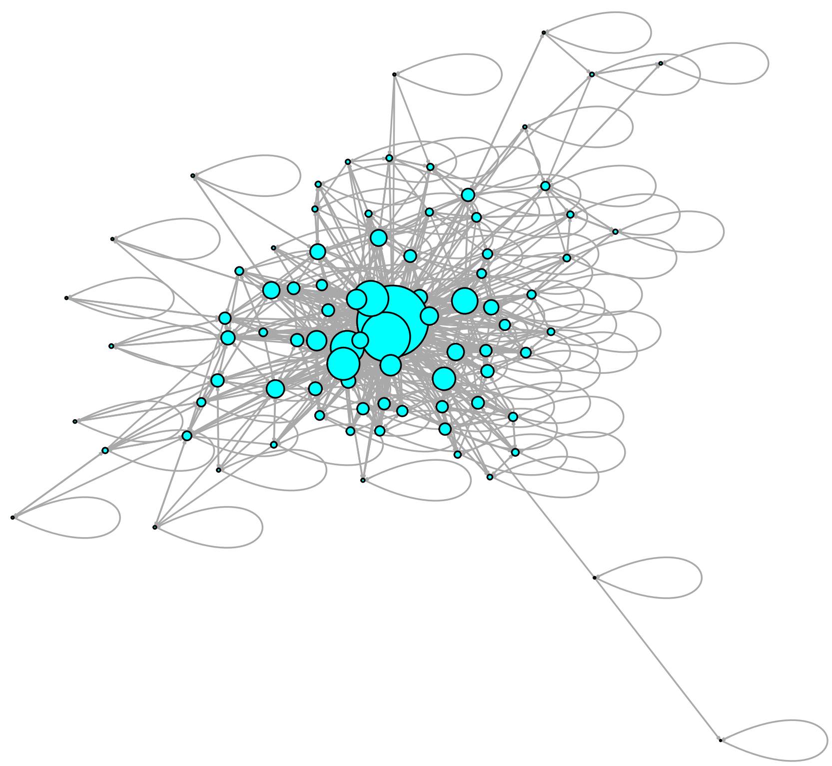


```
v2=paste("27", v1, sep="")
d_9394<-igraph::degree(g_9394)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_9394))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

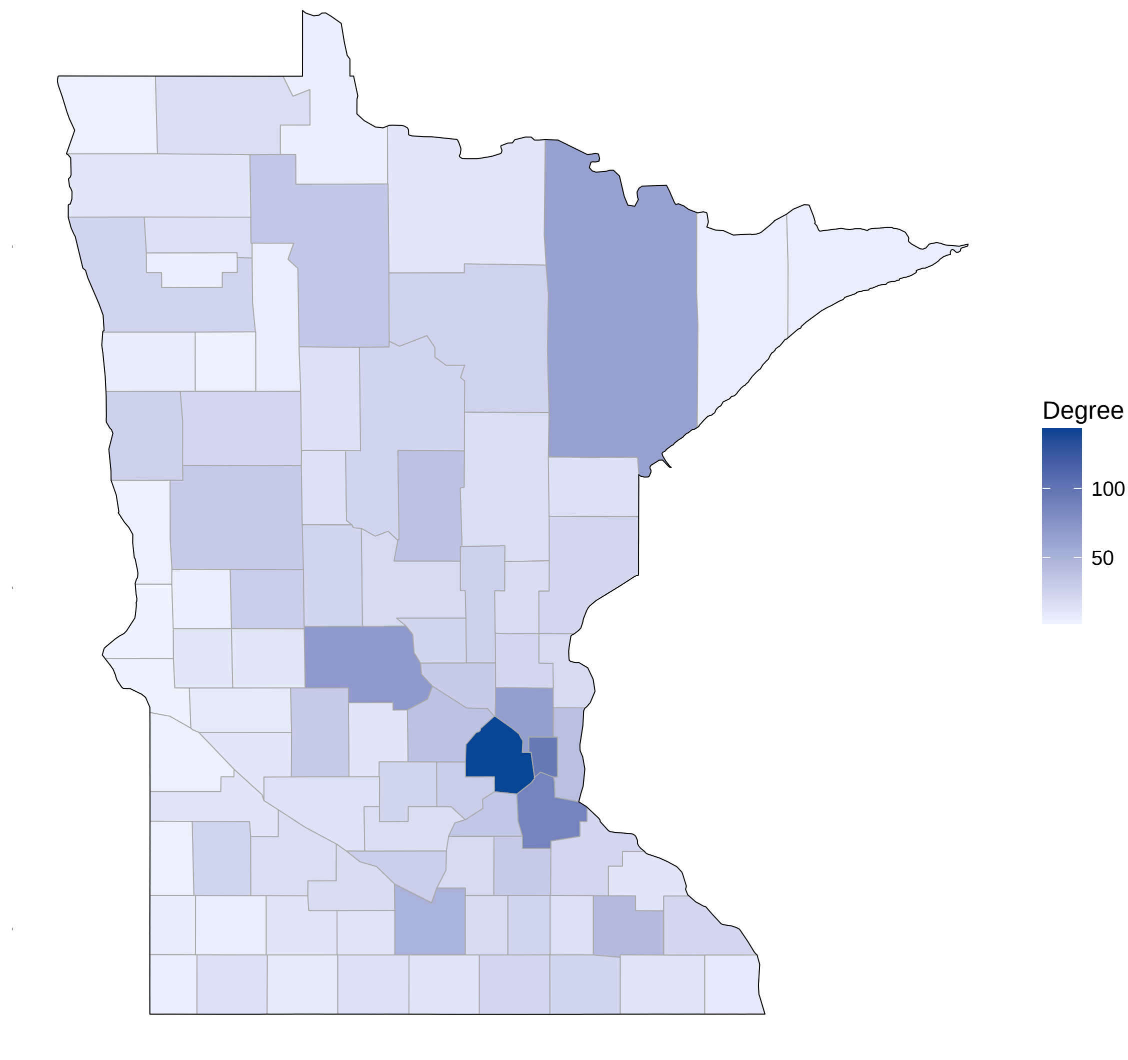


4.4 1994-1995

```
plot(g_9495, layout=layout.fruchterman.reingold(g_9495),
  vertex.size=igraph::degree(g_9495)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_9495), c(1,5)))
```

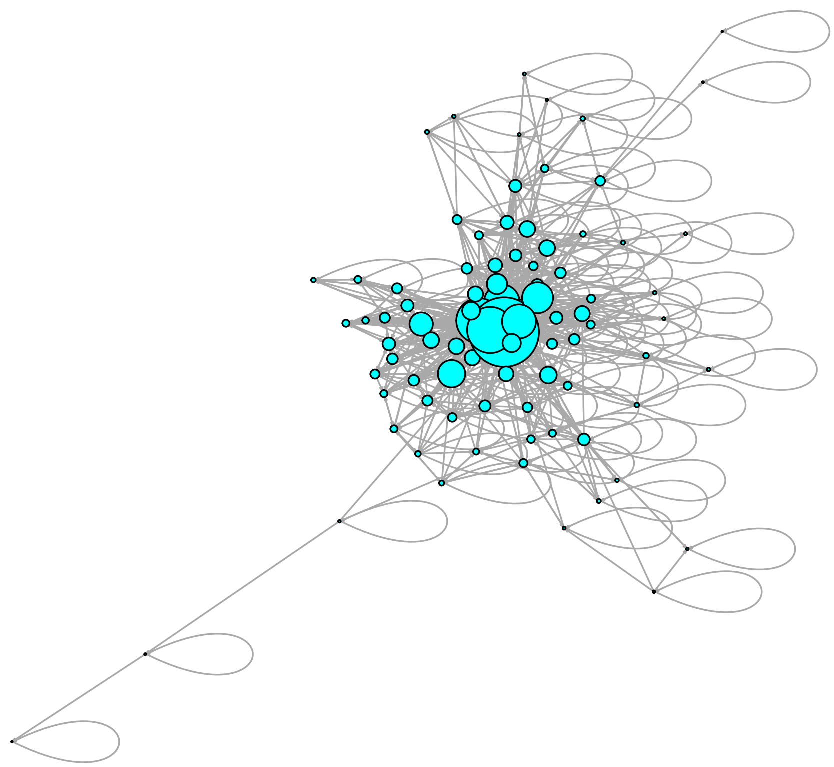


```
v2=paste("27", v1, sep="")
d_9495<-igraph::degree(g_9495)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_9495))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

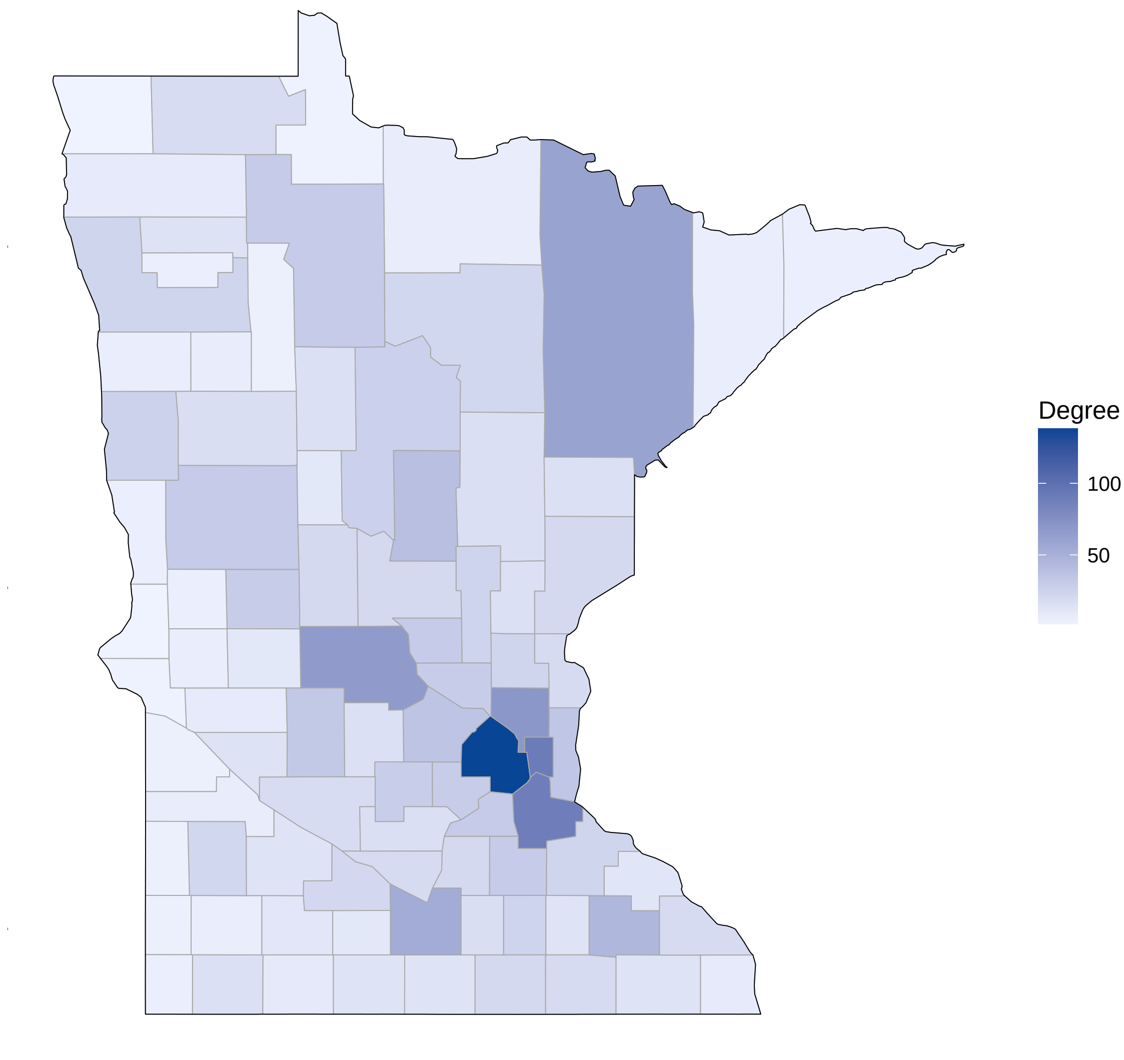


4.5 1995-1996

```
plot(g_9596, layout=layout.fruchterman.reingold(g_9596),
  vertex.size=igraph::degree(g_9596)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_9596), c(1,5)))
```

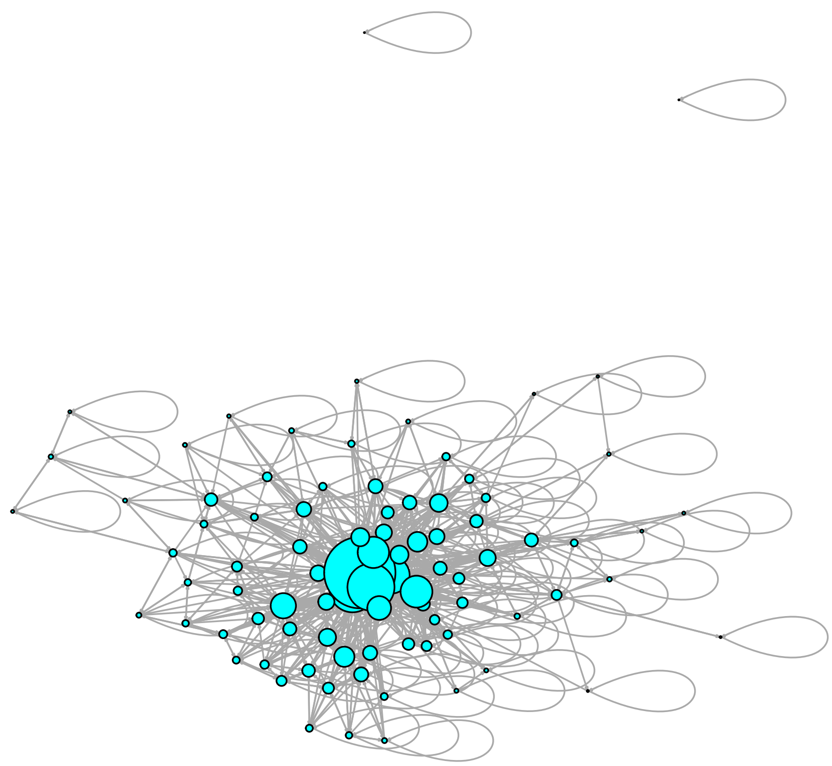


```
v2=paste("27", v1, sep="")
d_9596<-igraph::degree(g_9596)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_9596))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

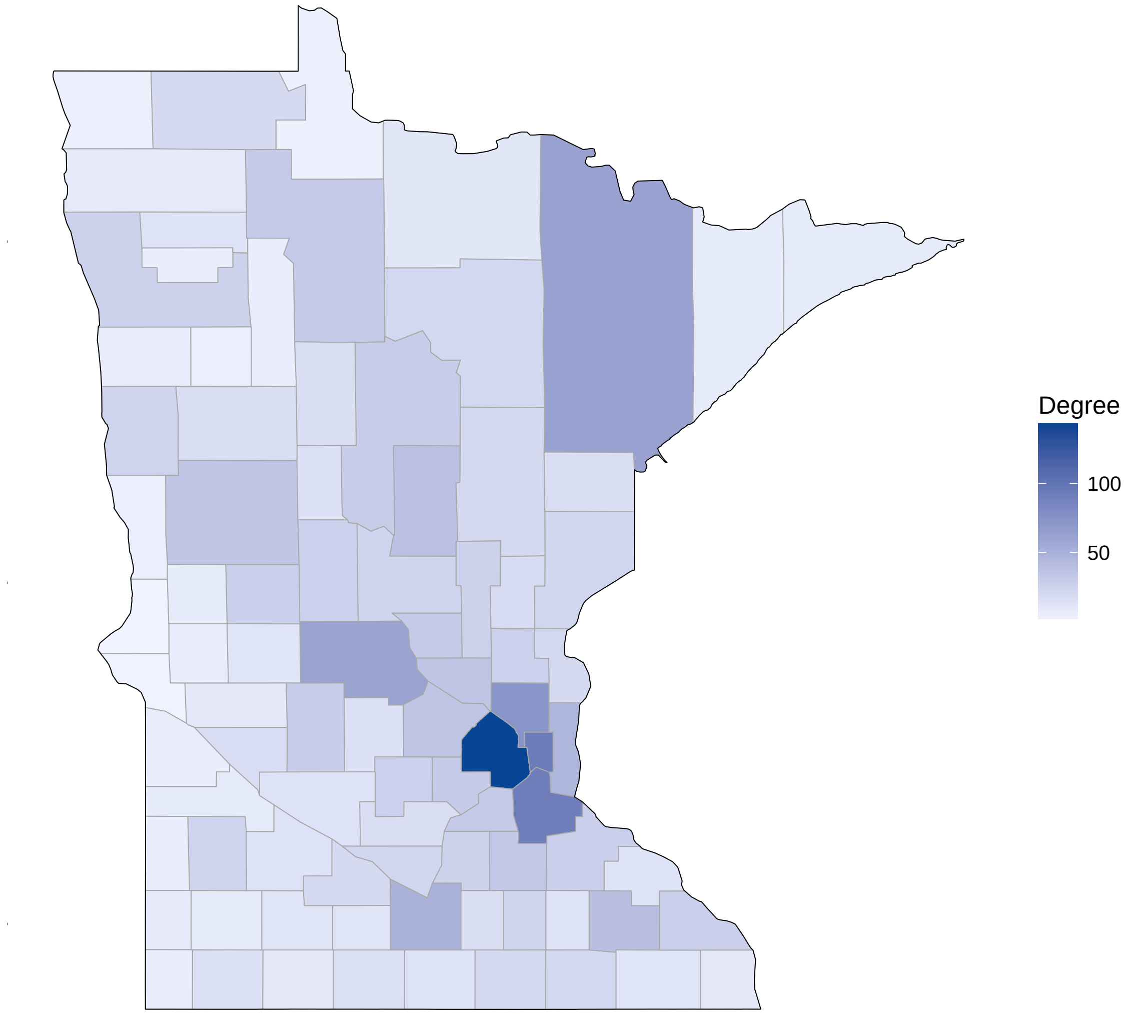


4.6 1996-1997

```
plot(g_9697, layout=layout.fruchterman.reingold(g_9697),
  vertex.size=igraph::degree(g_9697)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_9697), c(1,5)))
```

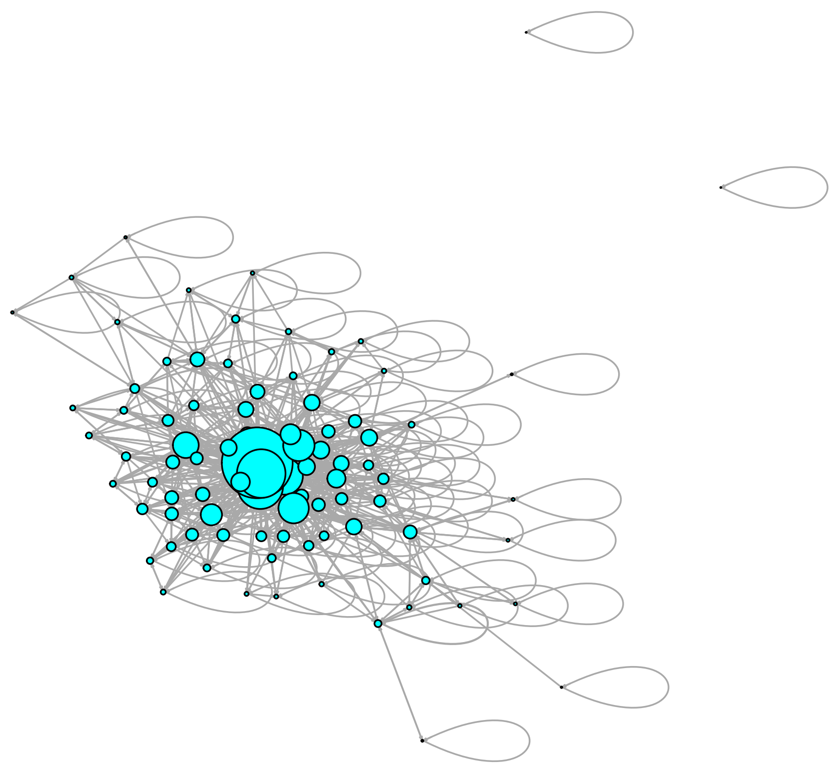


```
v2=paste("27", v1, sep="")
d_9697<-igraph::degree(g_9697)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_9697))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

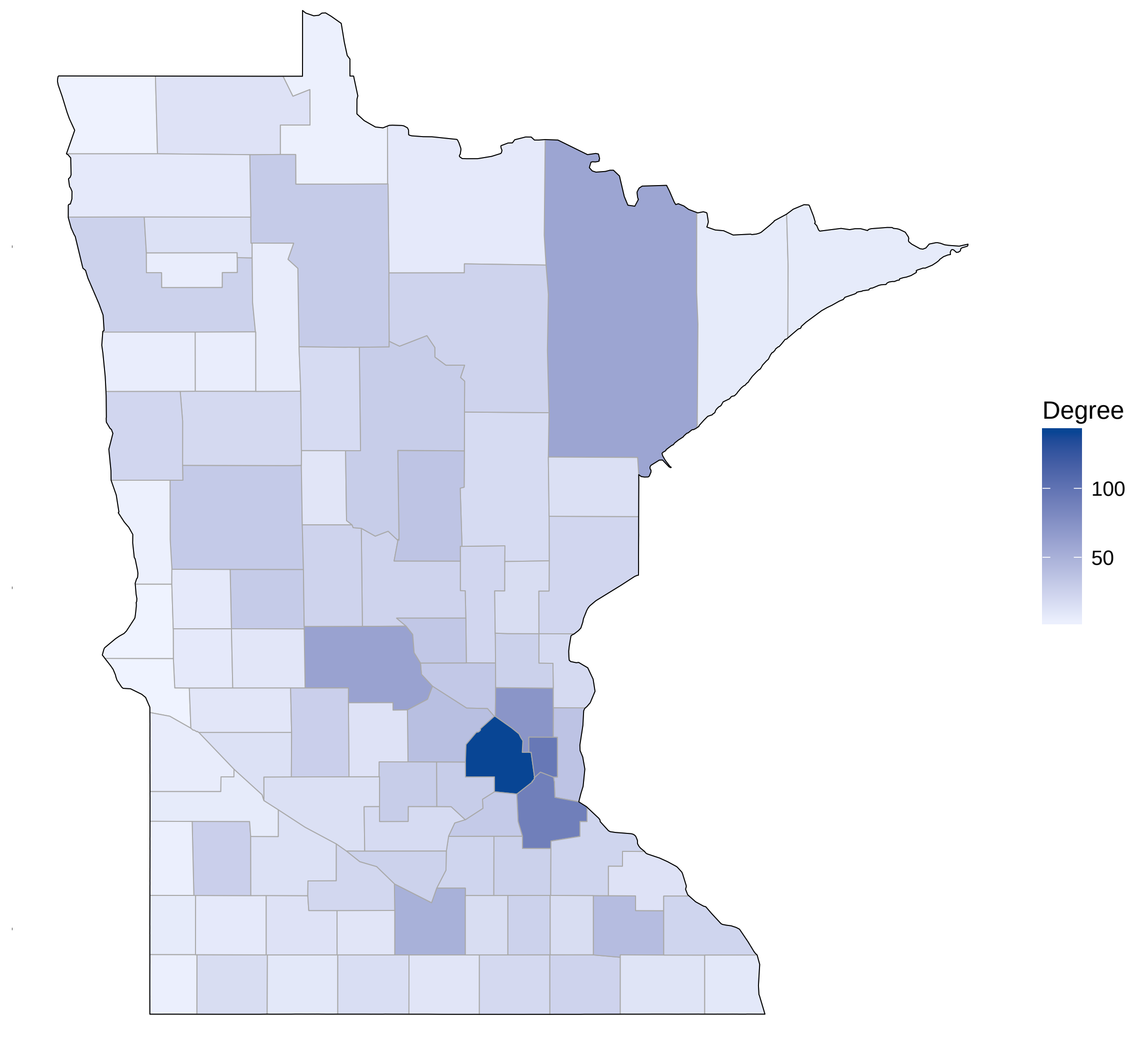


4.7 1997-1998

```
plot(g_9798, layout=layout.fruchterman.reingold(g_9798),
  vertex.size=igraph::degree(g_9798)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_9798), c(1,5)))
```

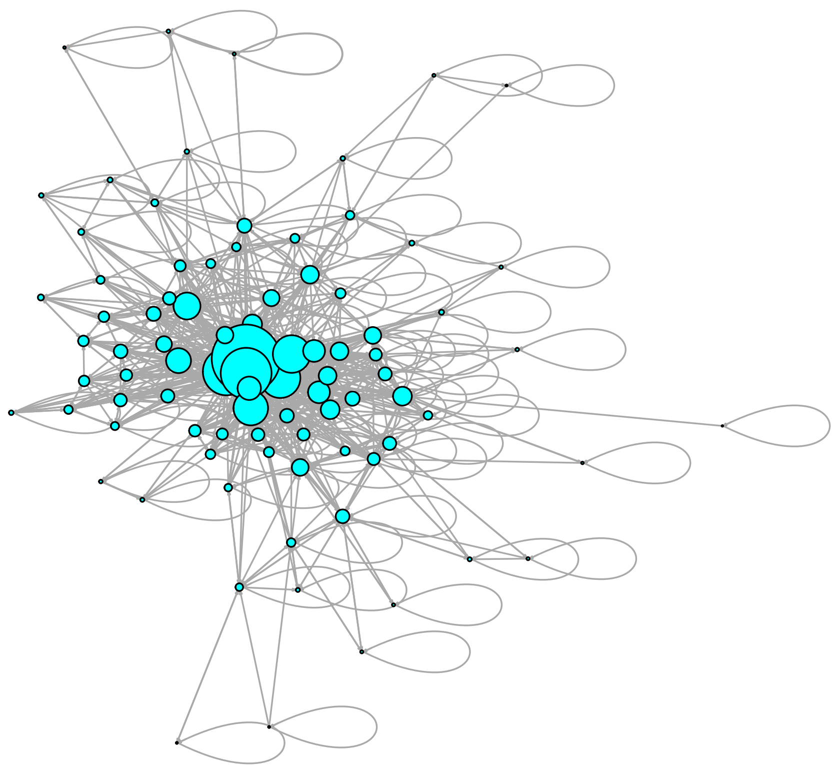


```
v2=paste("27", v1, sep="")
d_9798<-igraph::degree(g_9798)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_9798))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

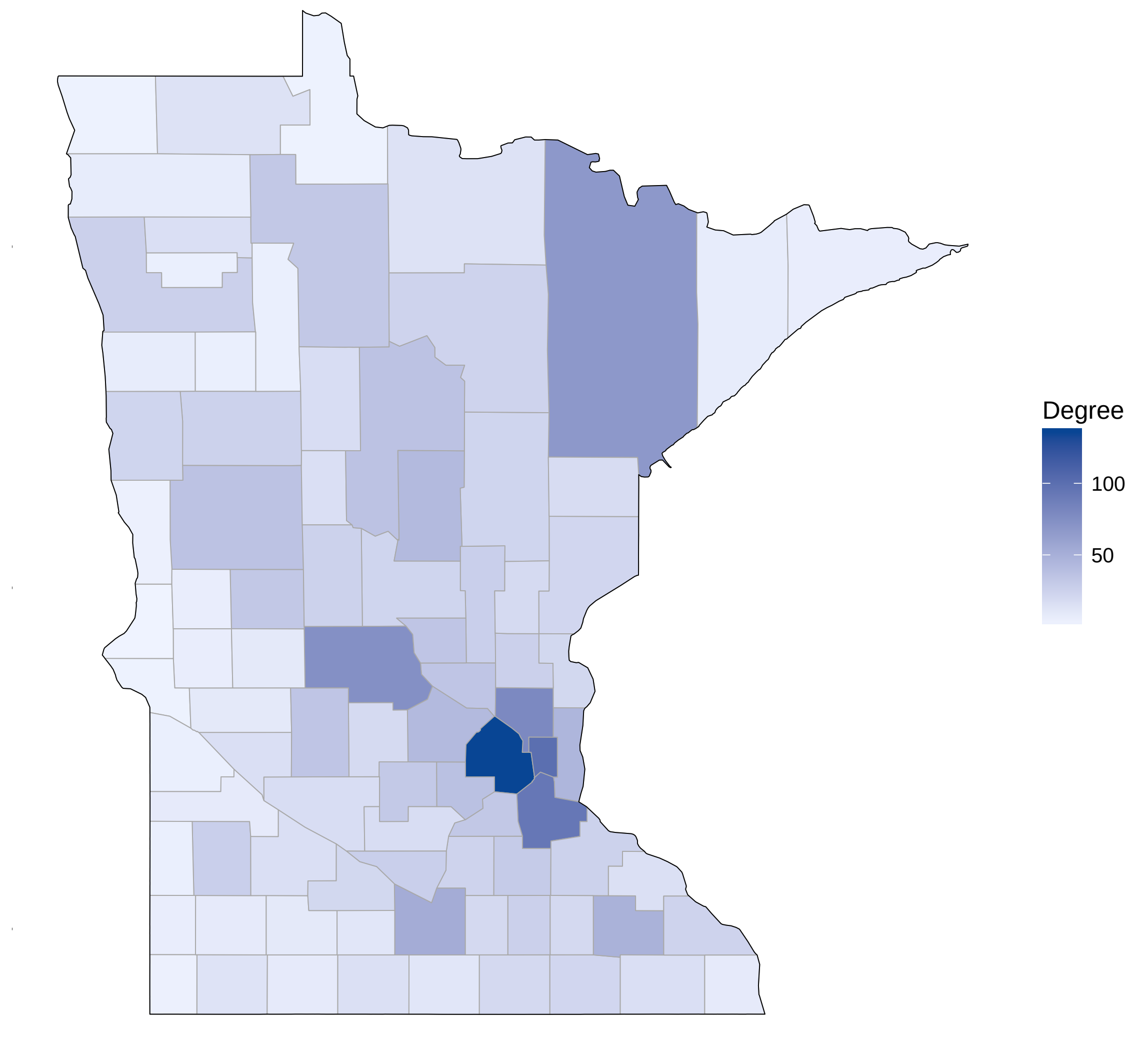


4.8 1998-1998

```
plot(g_9899, layout=layout.fruchterman.reingold(g_9899),
  vertex.size=igraph::degree(g_9899)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_9899), c(1,5)))
```

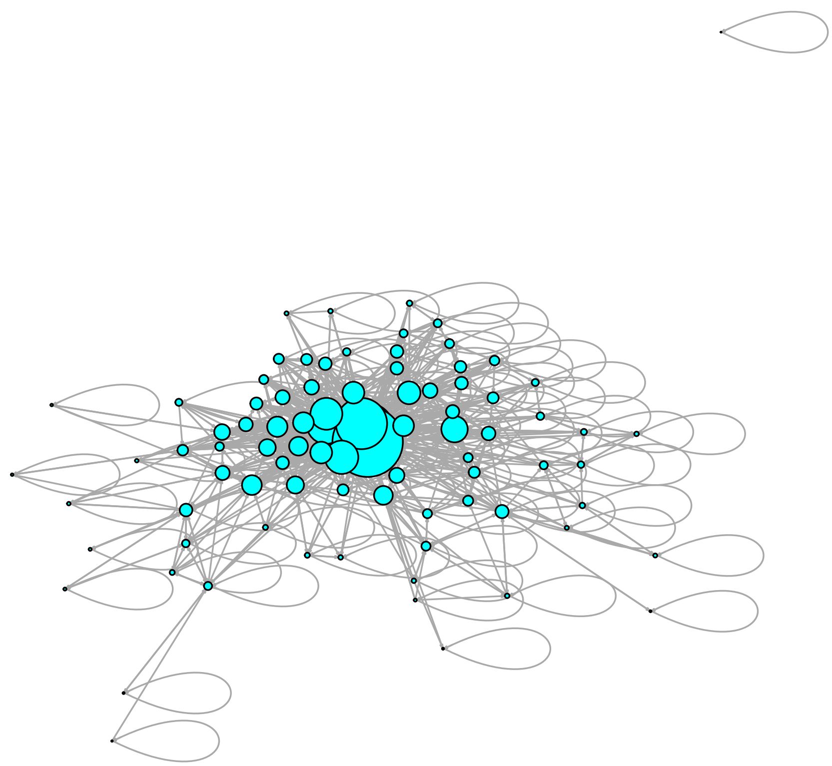


```
v2=paste("27", v1, sep="")
d_9899<-igraph::degree(g_9899)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_9899))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

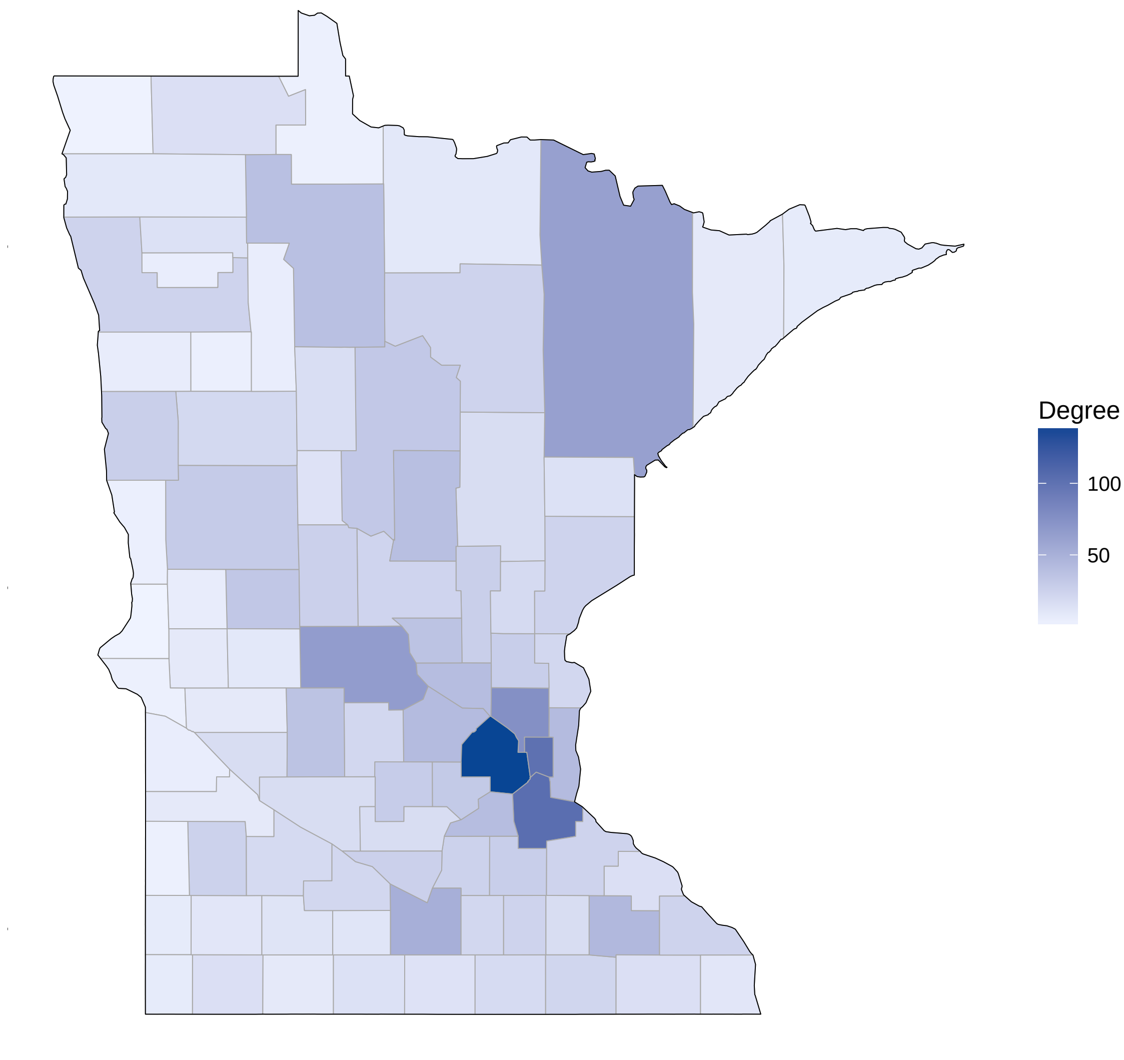


4.9 2000-2001

```
plot(g_0001, layout=layout.fruchterman.reingold(g_0001),
  vertex.size=igraph::degree(g_0001)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0001), c(1,5)))
```

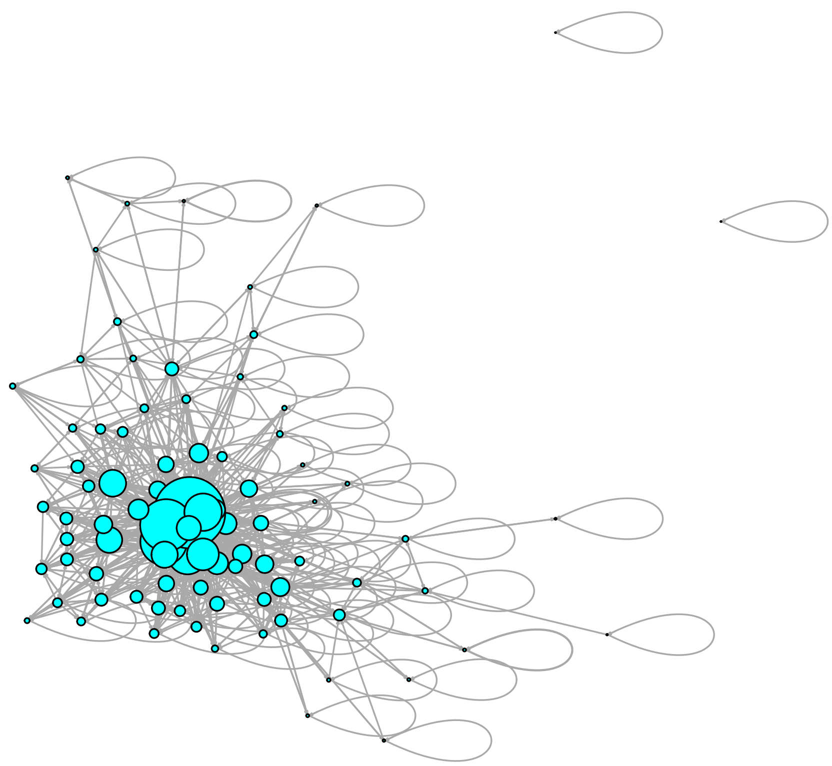


```
v2=paste("27", v1, sep="")
d_0001<-igraph::degree(g_0001)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0001))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

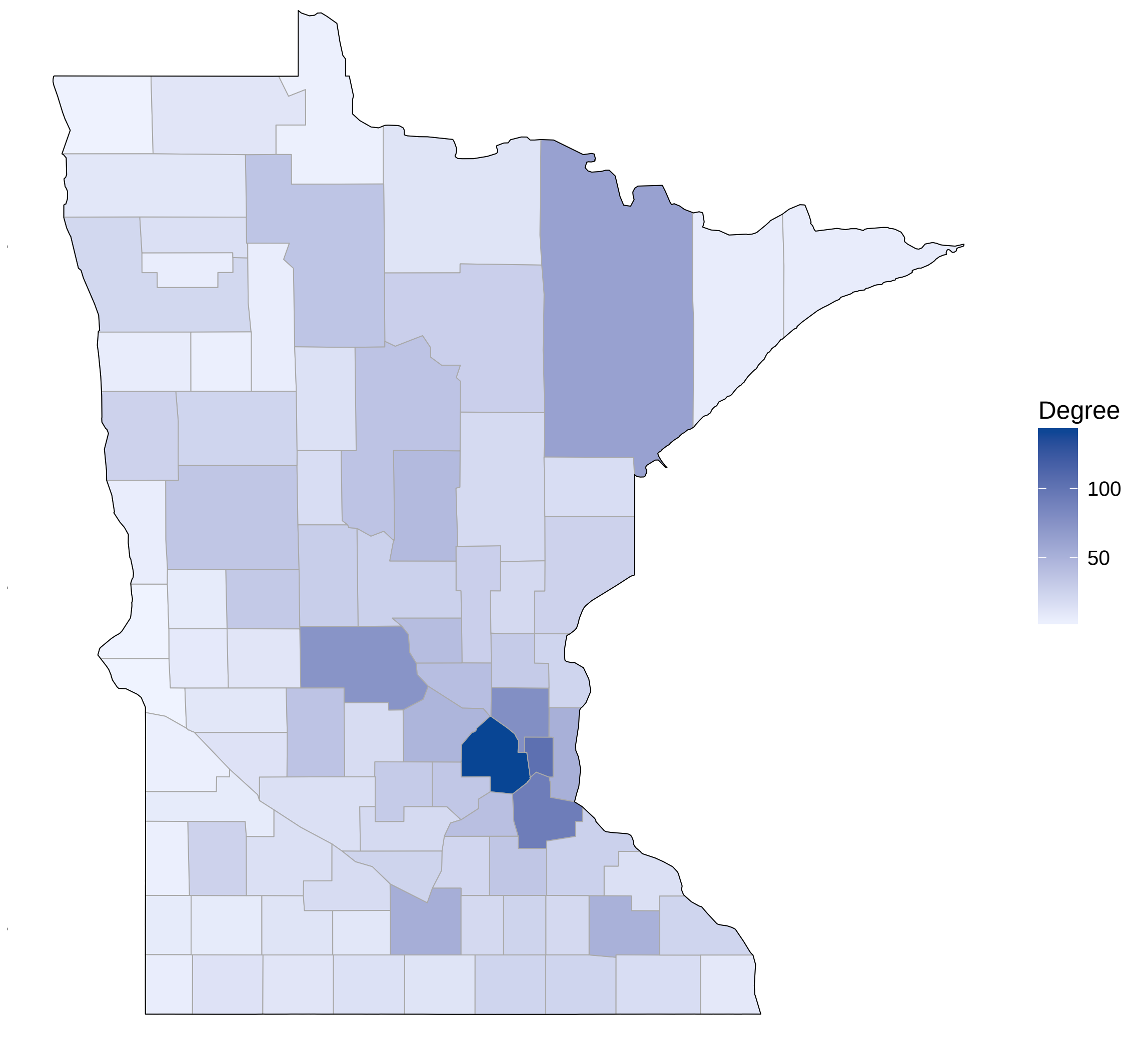


4.10 2001-2002

```
plot(g_0102, layout=layout.fruchterman.reingold(g_0102),
  vertex.size=igraph::degree(g_0102)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0102), c(1,5)))
```

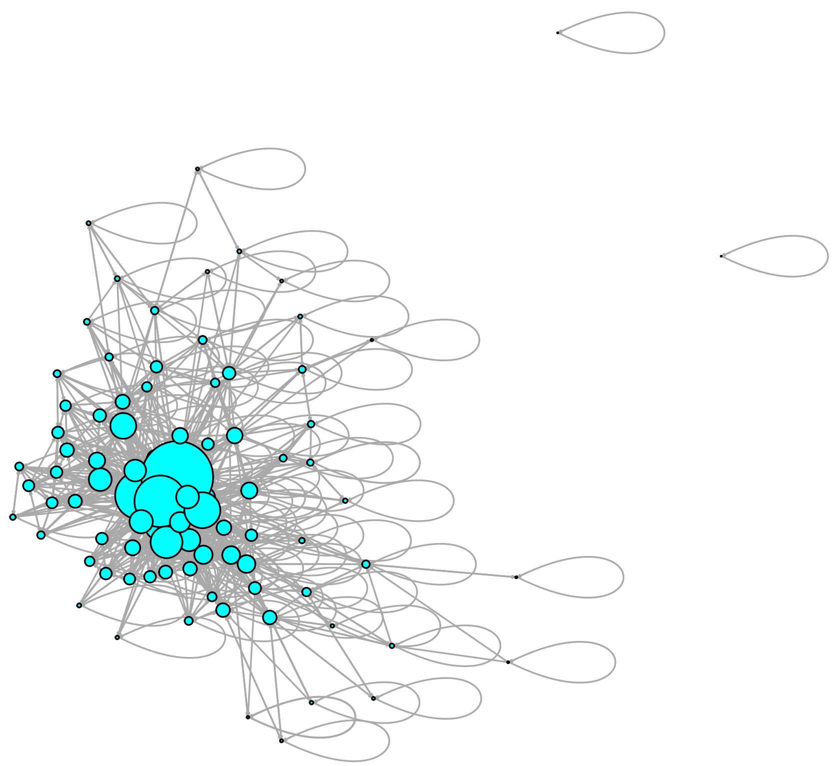


```
v2=paste("27", v1, sep="")
d_0102<-igraph::degree(g_0102)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0102))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

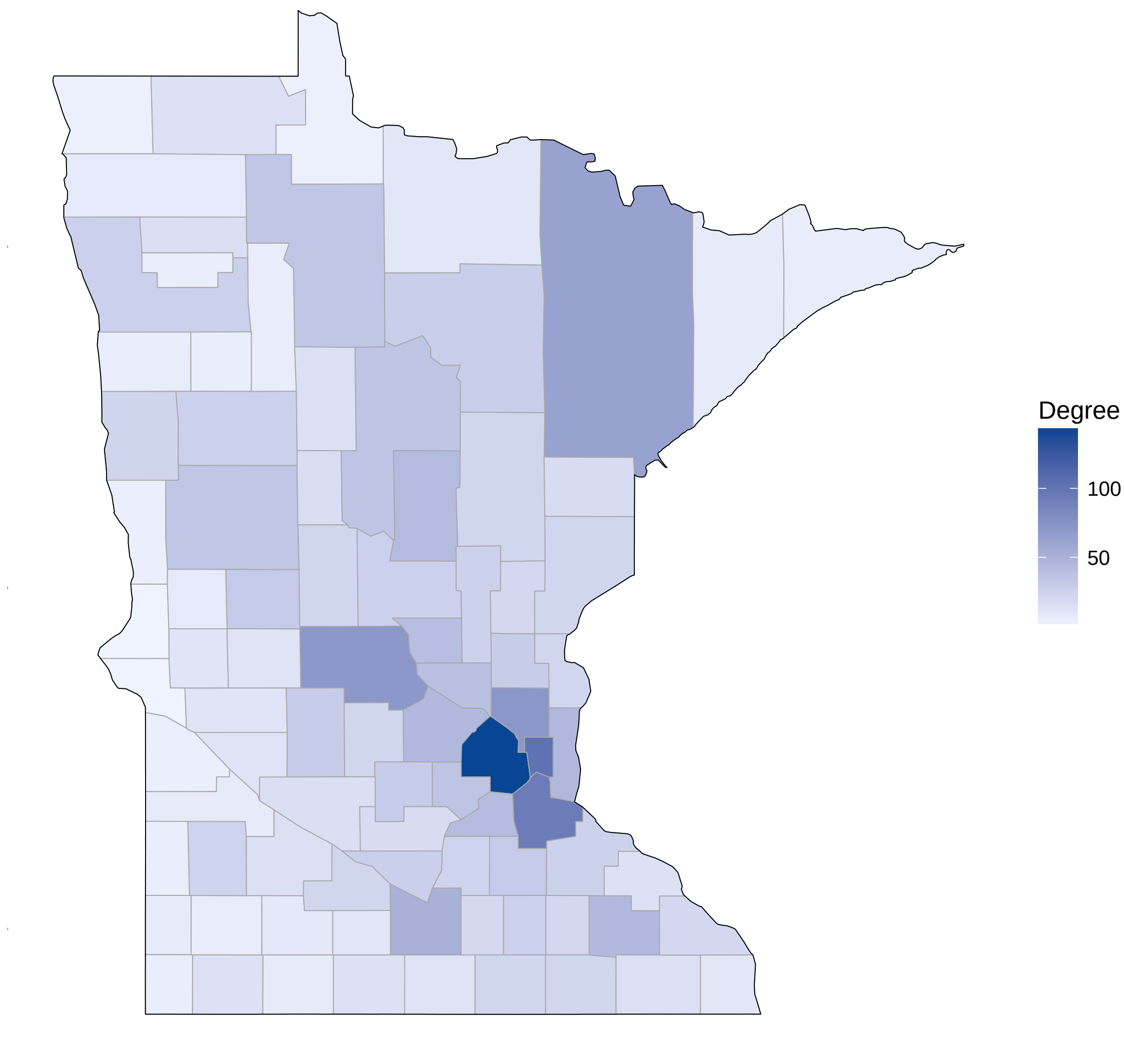


4.11 2002-2003

```
plot(g_0203, layout=layout.fruchterman.reingold(g_0203),
  vertex.size=igraph::degree(g_0203)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0203), c(1,5)))
```

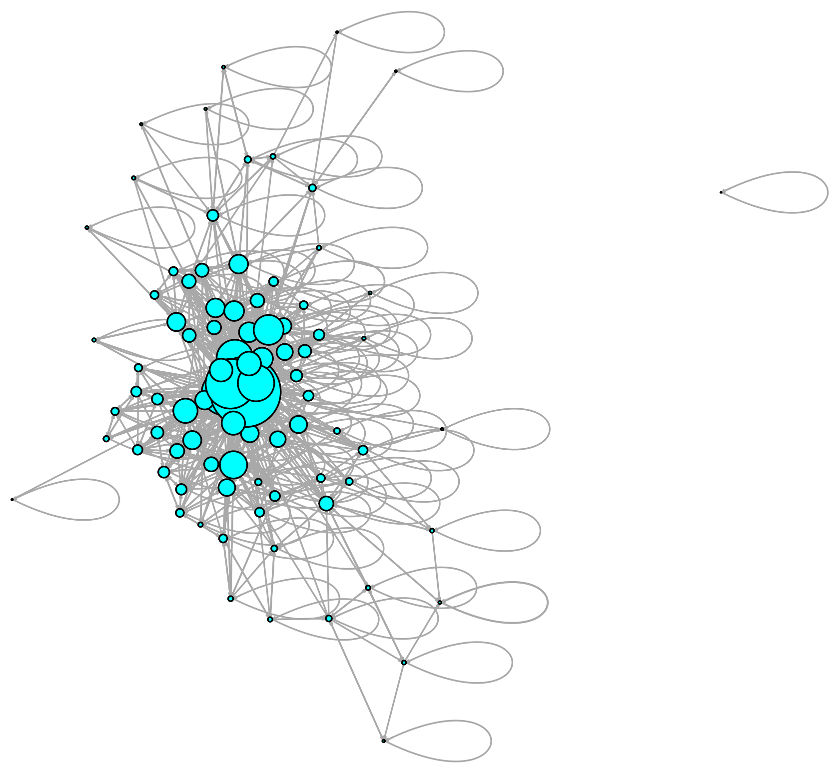


```
v2=paste("27", v1, sep="")
d_0203<-igraph::degree(g_0203)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0203))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

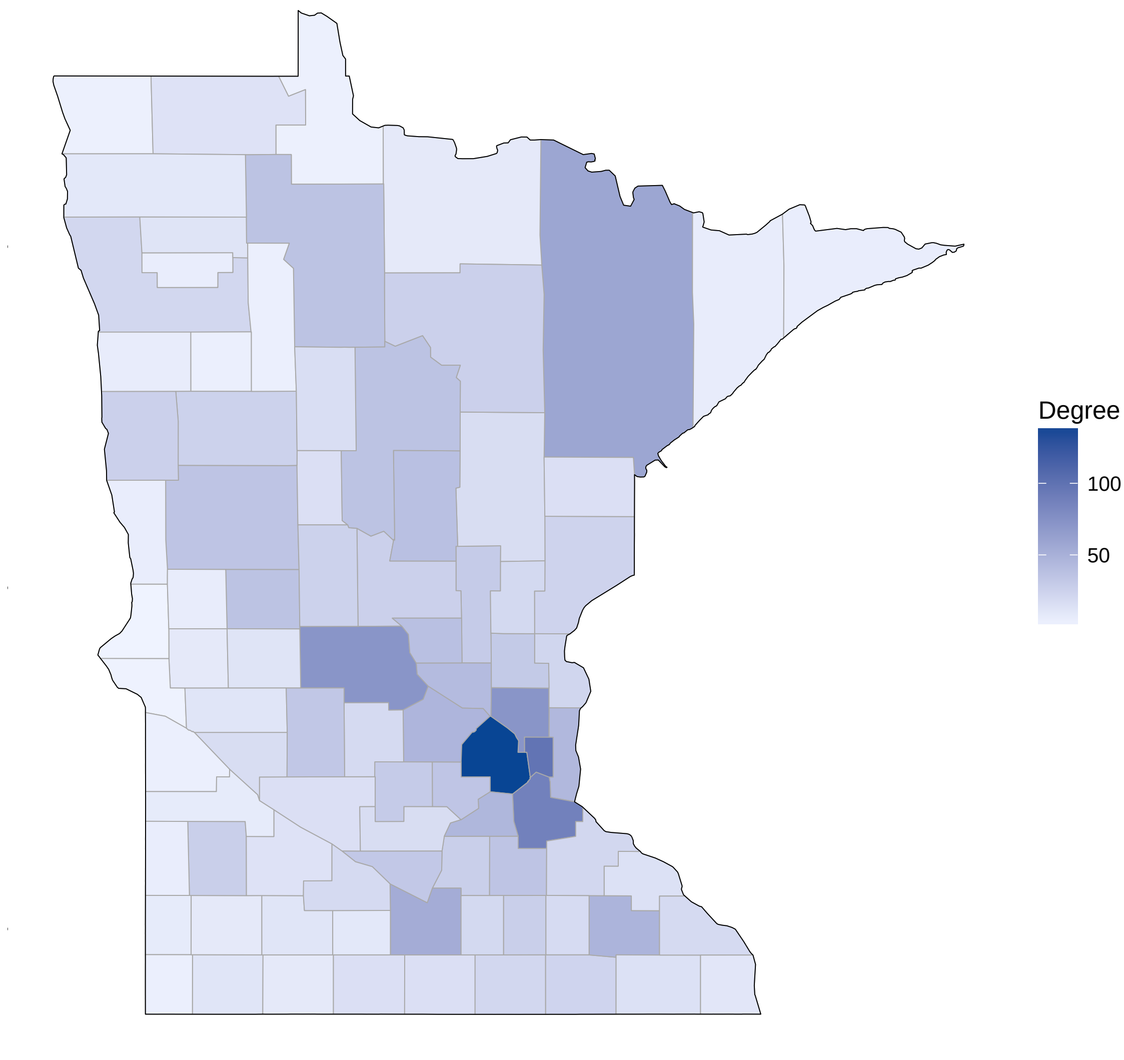


4.12 2003-2004

```
plot(g_0304, layout=layout.fruchterman.reingold(g_0304),
  vertex.size=igraph::degree(g_0304)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0304), c(1,5)))
```

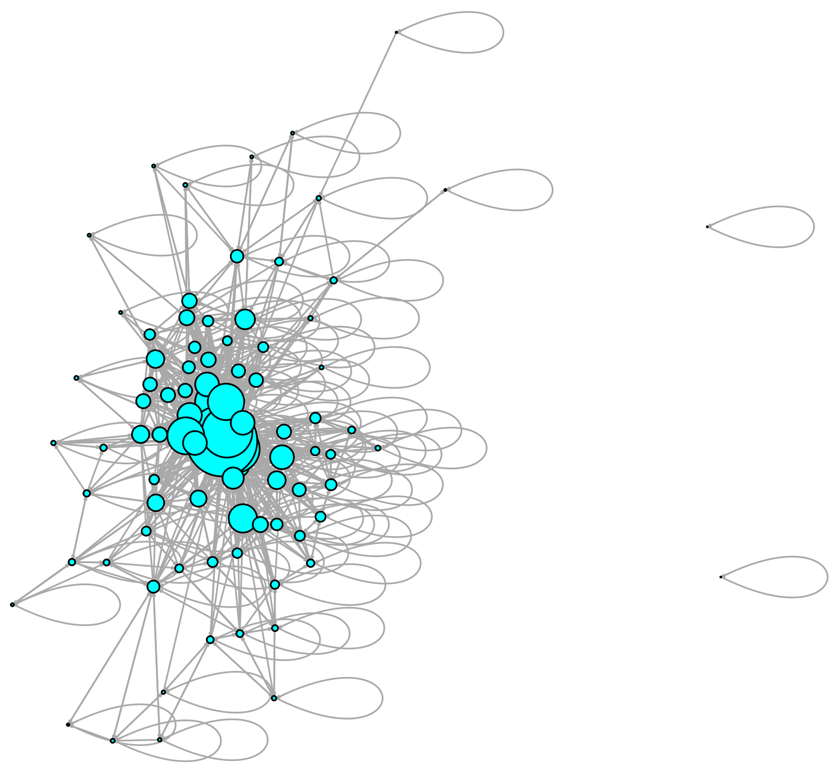


```r
v2=paste("27", v1, sep="")
d_0304<-igraph::degree(g_0304)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0304))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

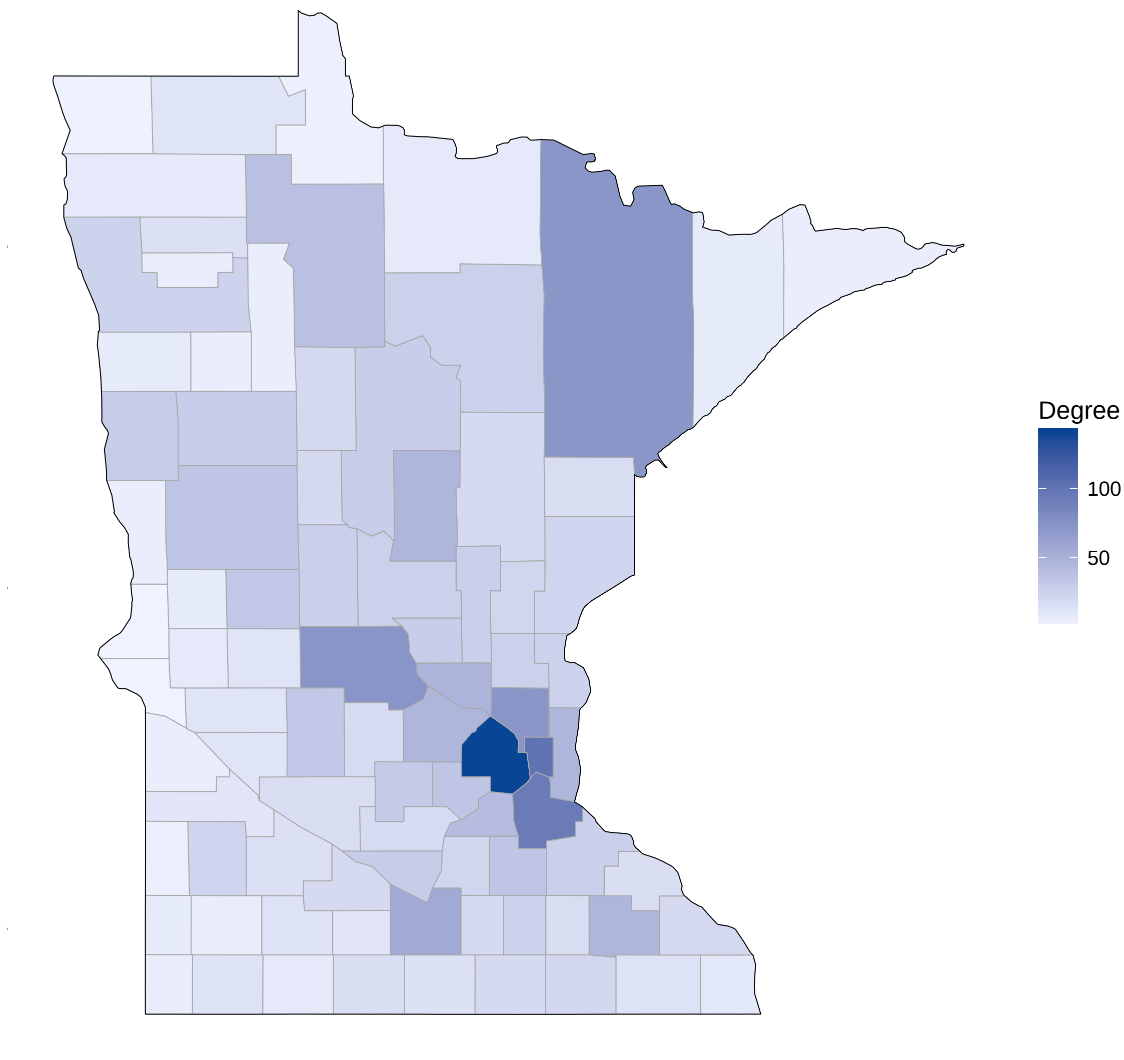


4.13 2004-2005

```
plot(g_0405, layout=layout.fruchterman.reingold(g_0405),
  vertex.size=igraph::degree(g_0405)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0405), c(1,5)))
```

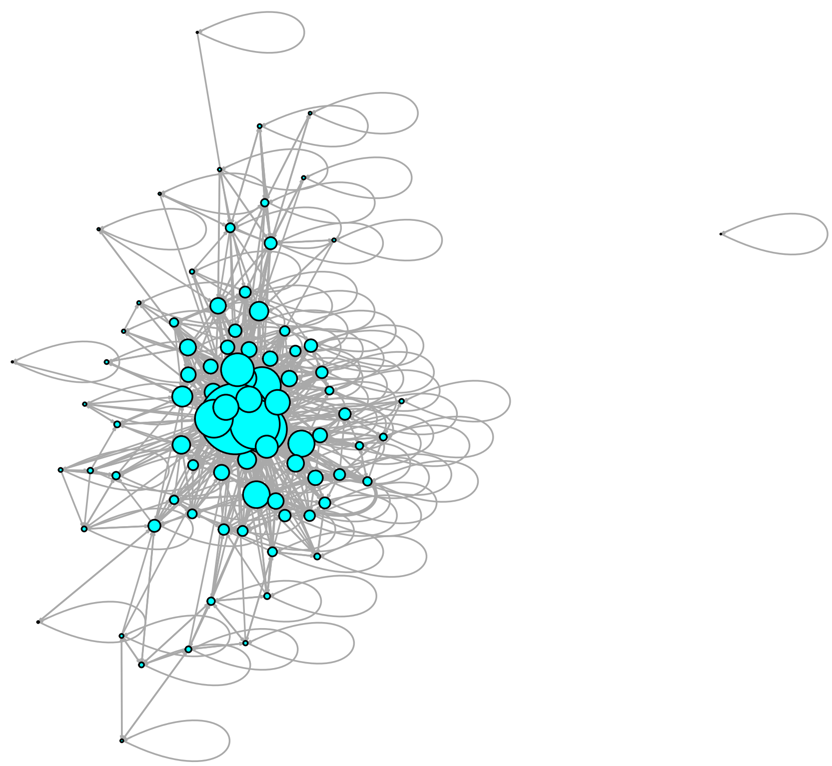


```
v2=paste("27", v1, sep="")
d_0405<-igraph::degree(g_0405)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0405))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

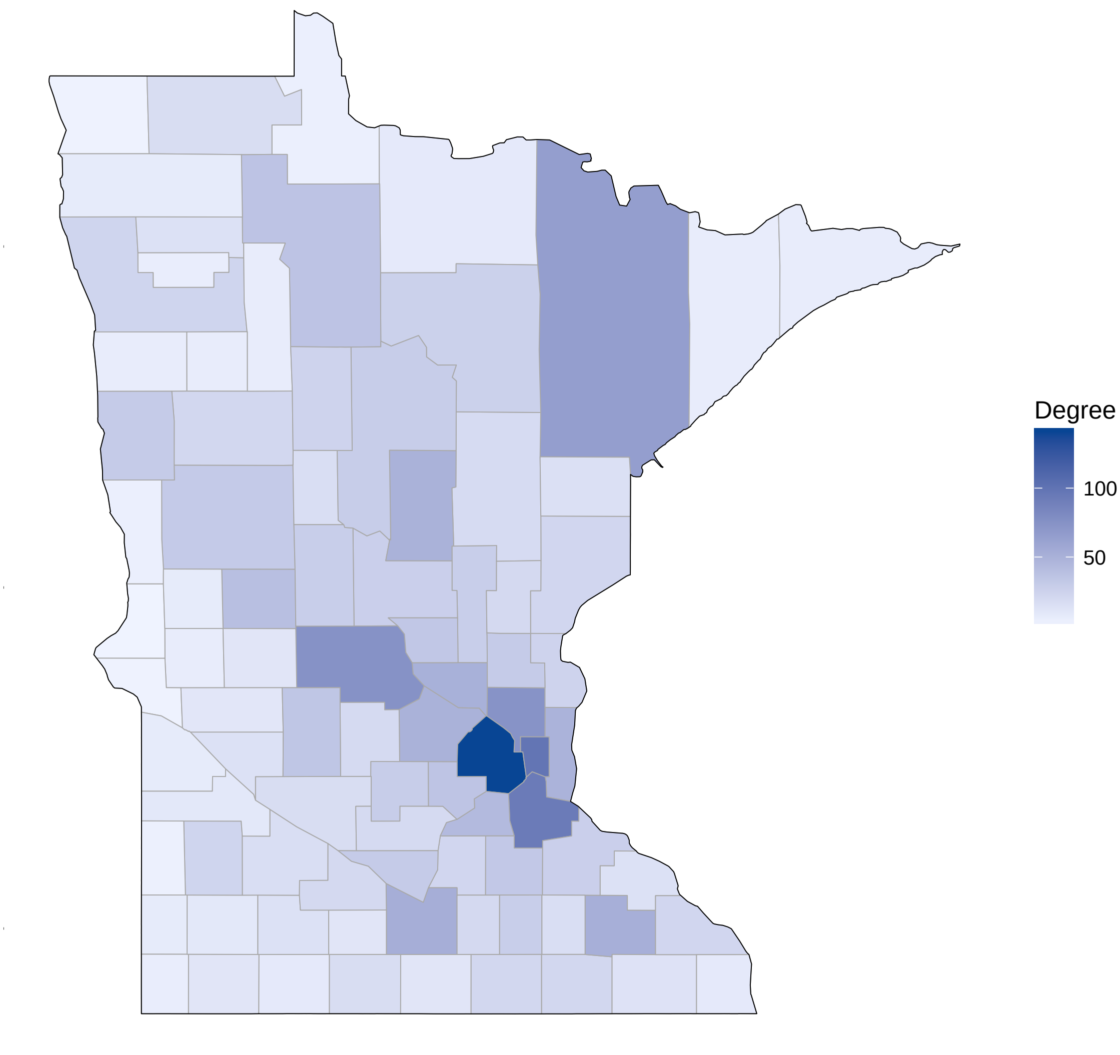


4.14 2005-2006

```
plot(g_0506, layout=layout.fruchterman.reingold(g_0506),
  vertex.size=igraph::degree(g_0506)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0506), c(1,5)))
```

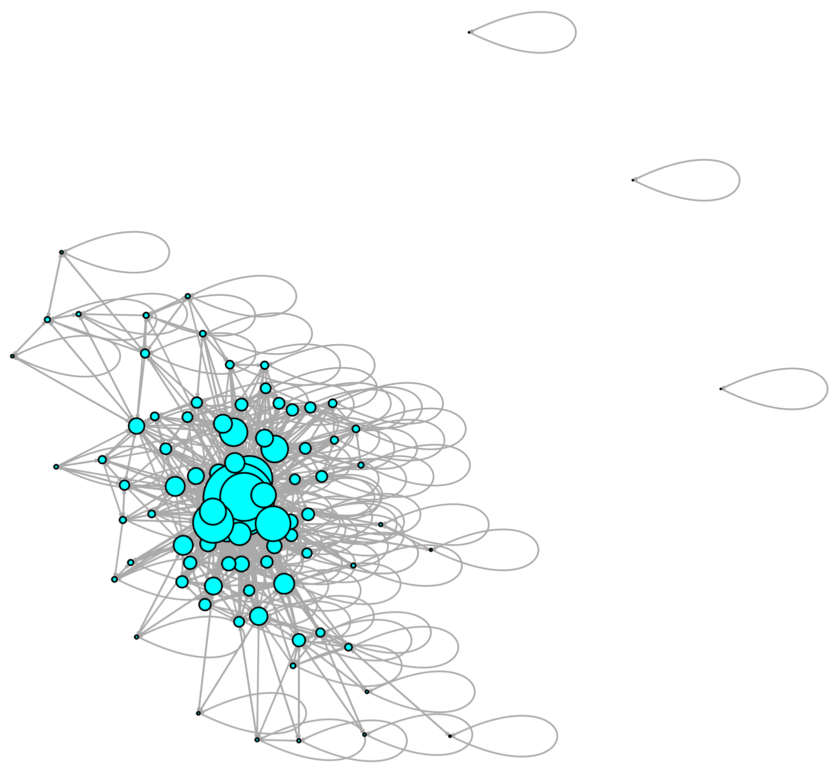


```
v2=paste("27", v1, sep="")
d_0506<-igraph::degree(g_0506)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0506))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

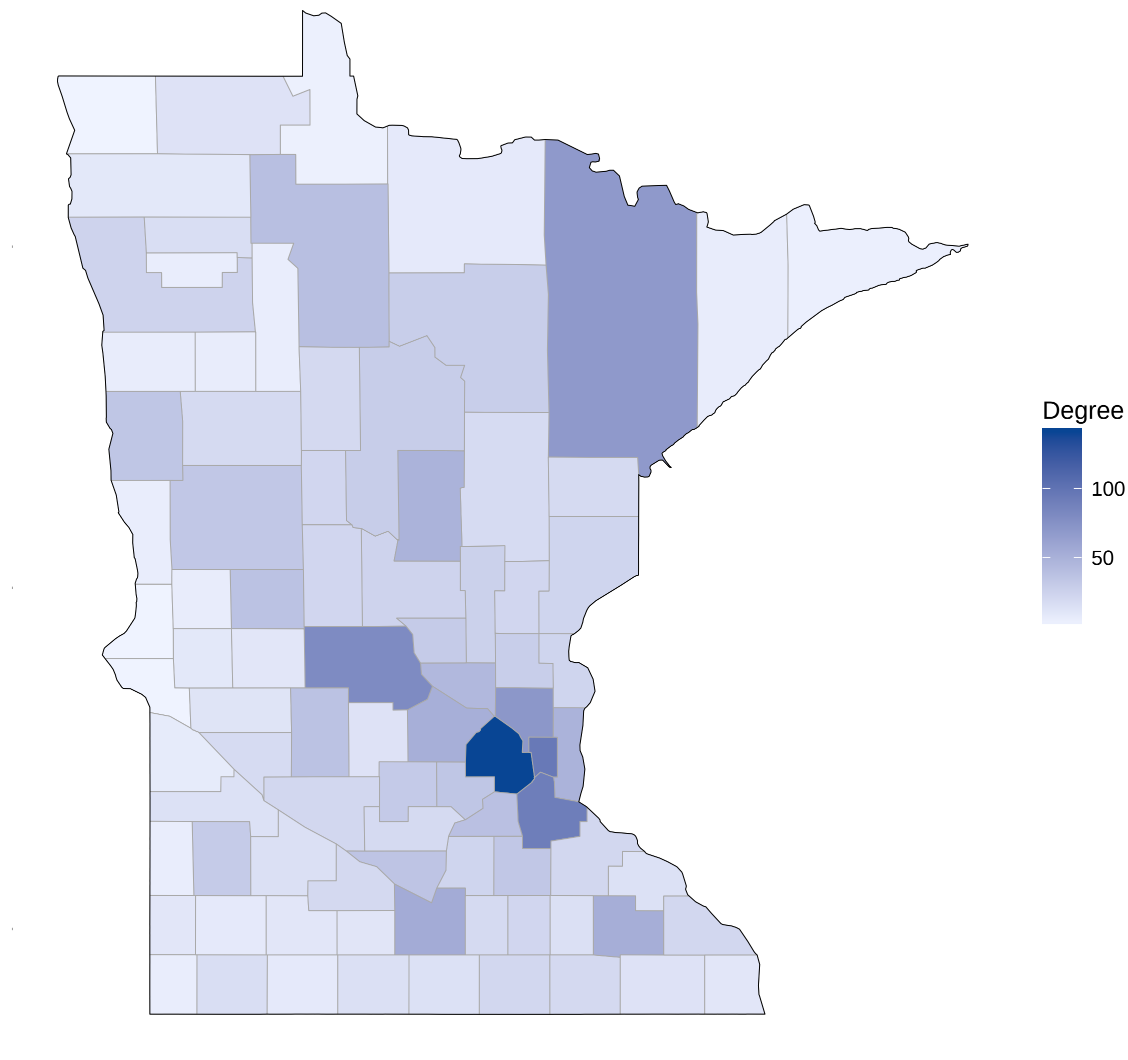


4.15 2006-2007

```
plot(g_0607, layout=layout.fruchterman.reingold(g_0607),
  vertex.size=igraph::degree(g_0607)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0607), c(1,5)))
```

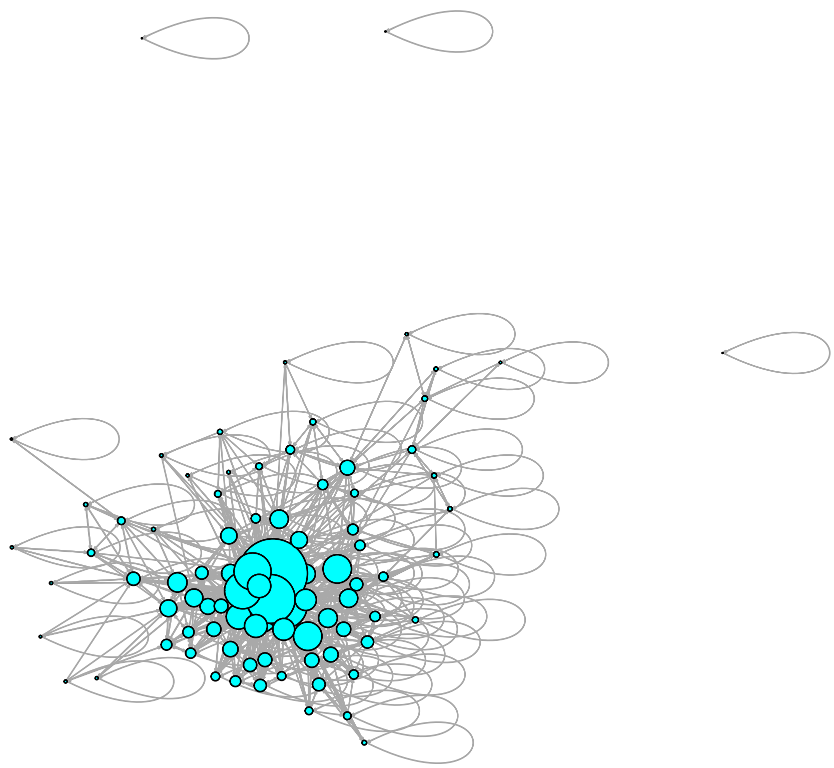


```
v2=paste("27", v1, sep="")
d_0607<-igraph::degree(g_0607)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0607))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

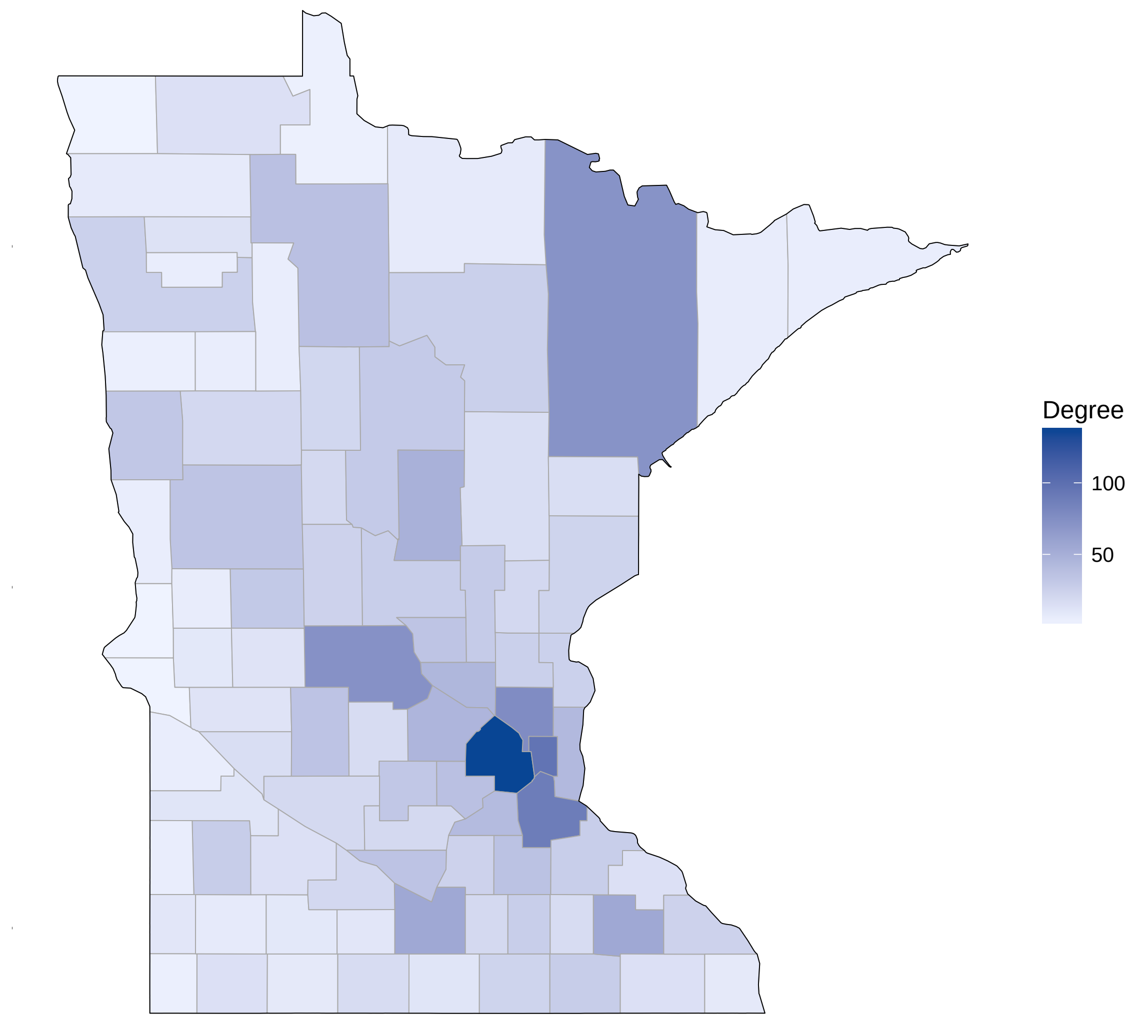


4.16 2007-2008

```
plot(g_0708, layout=layout.fruchterman.reingold(g_0708),
  vertex.size=igraph::degree(g_0708)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0708), c(1,5)))
```

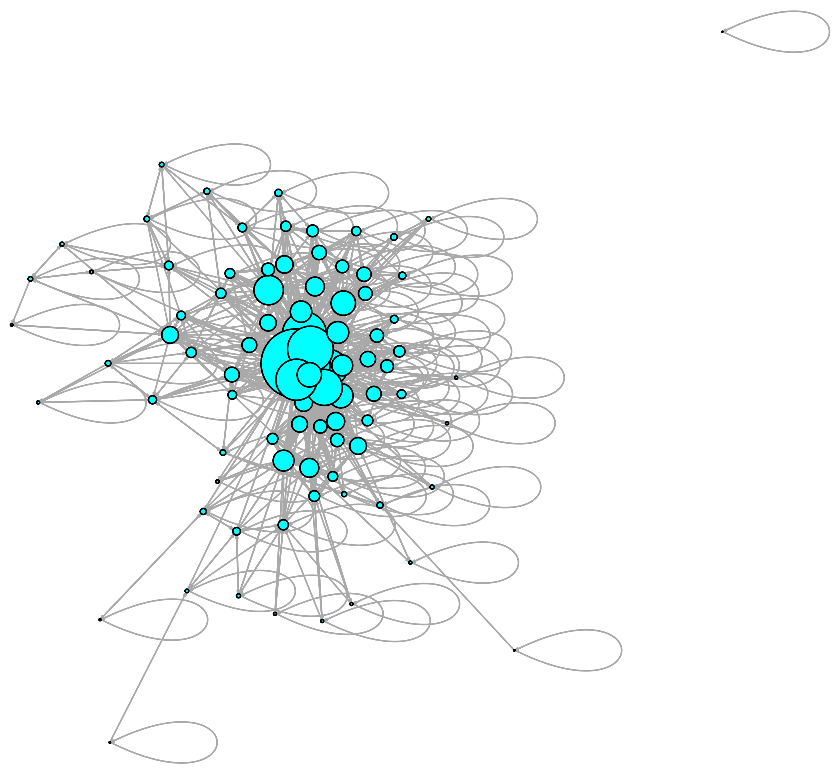


```
v2=paste("27", v1, sep="")
d_0708<-igraph::degree(g_0708)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0708))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

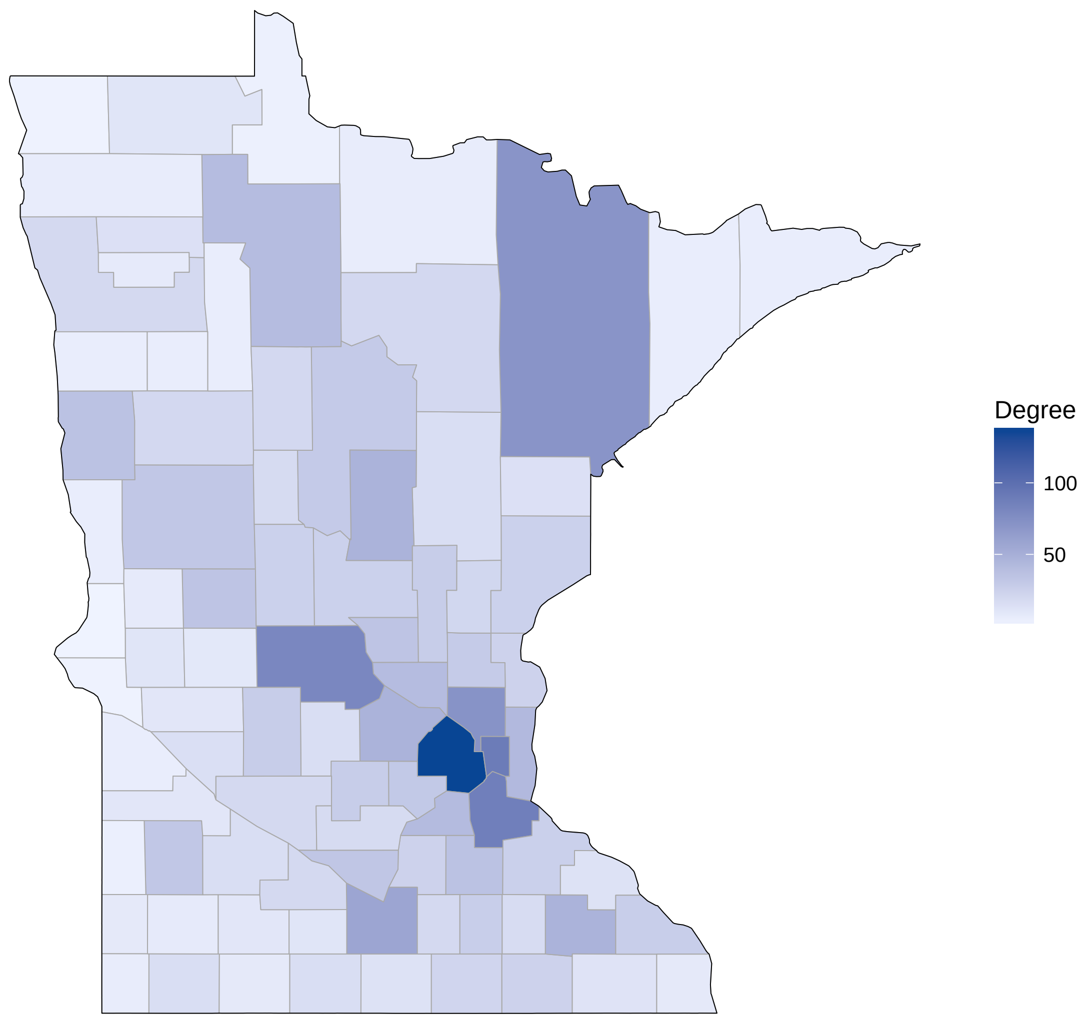


4.17 2008-2009

```
plot(g_0809, layout=layout.fruchterman.reingold(g_0809),
  vertex.size=igraph::degree(g_0809)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0809), c(1,5)))
```

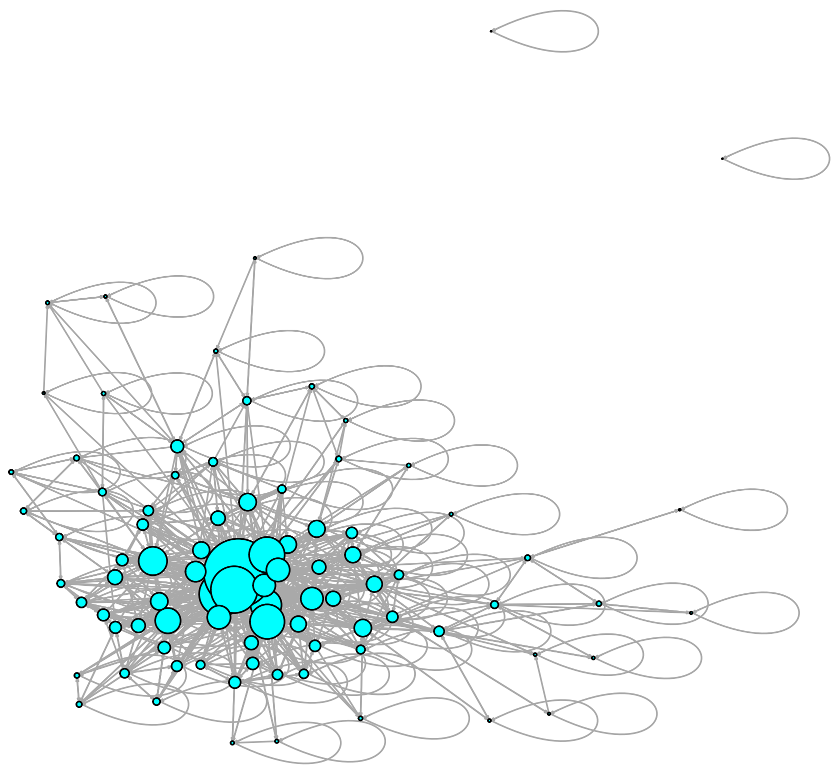


```
v2=paste("27", v1, sep="")
d_0809<-igraph::degree(g_0809)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0809))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

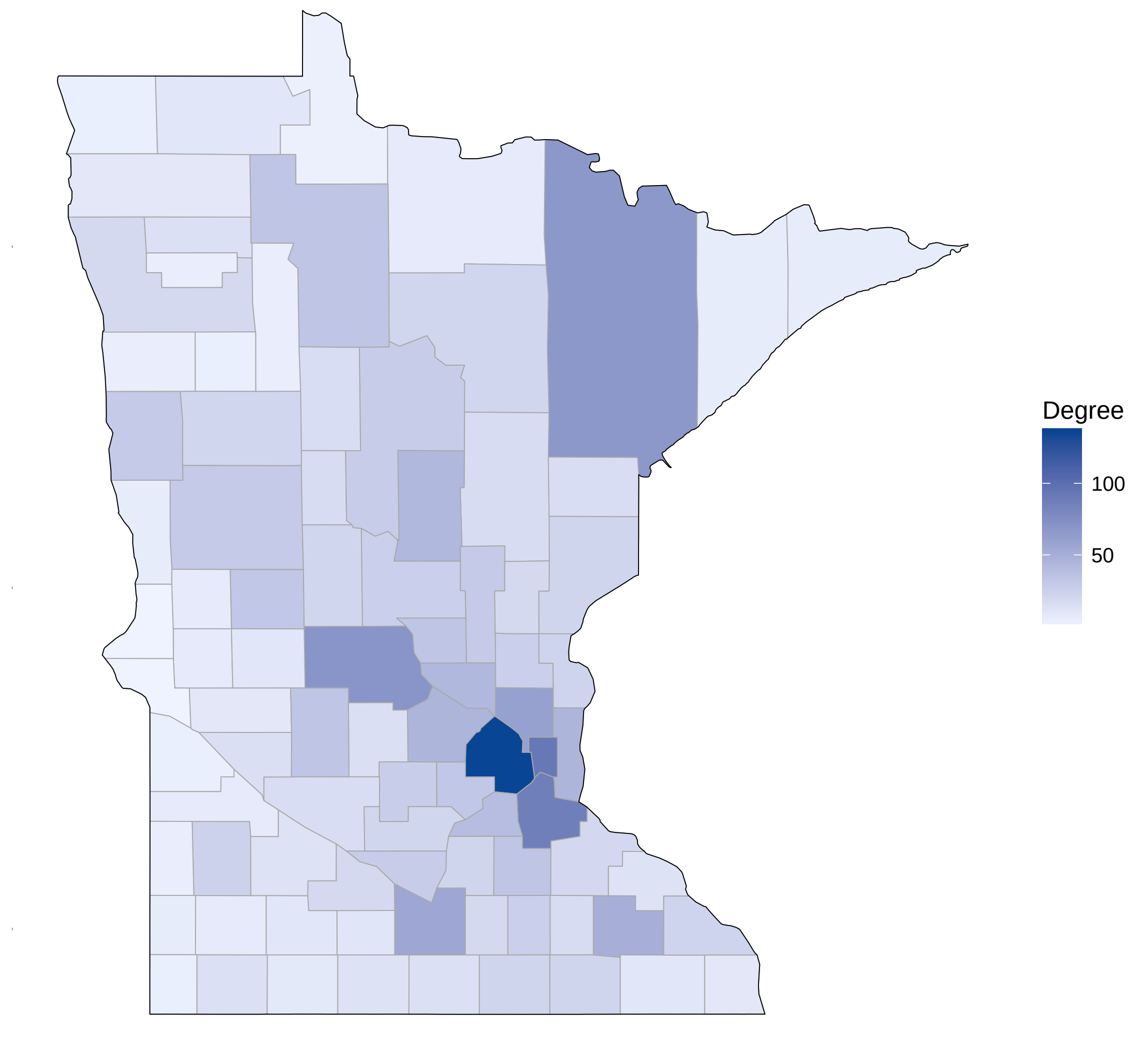


4.18 2009-2010

```
plot(g_0910, layout=layout.fruchterman.reingold(g_0910),
  vertex.size=igraph::degree(g_0910)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_0910), c(1,5)))
```

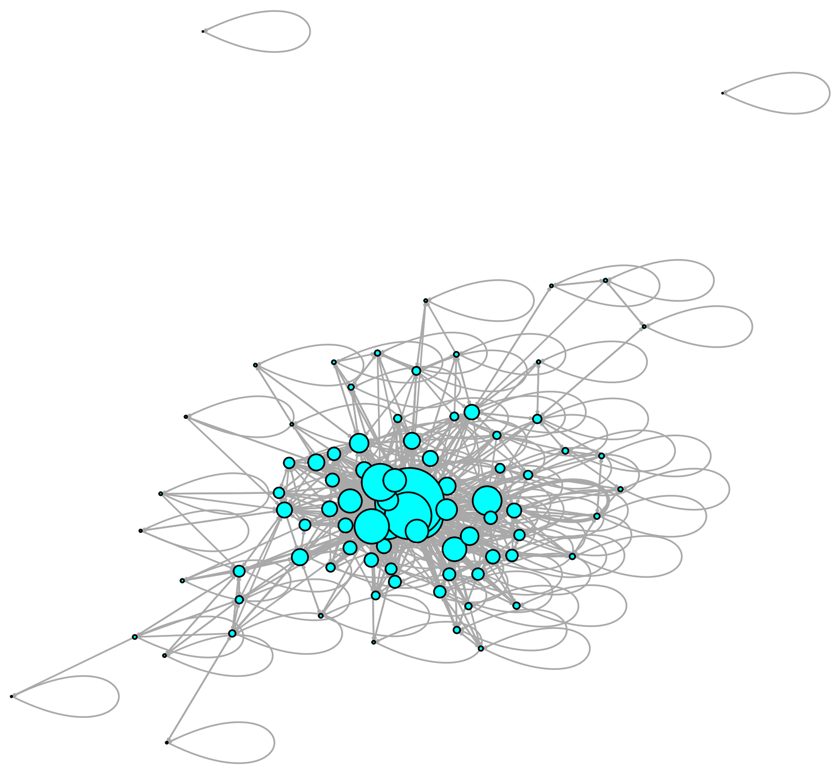


```
v2=paste("27", v1, sep="")
d_0910<-igraph::degree(g_0910)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_0910))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```

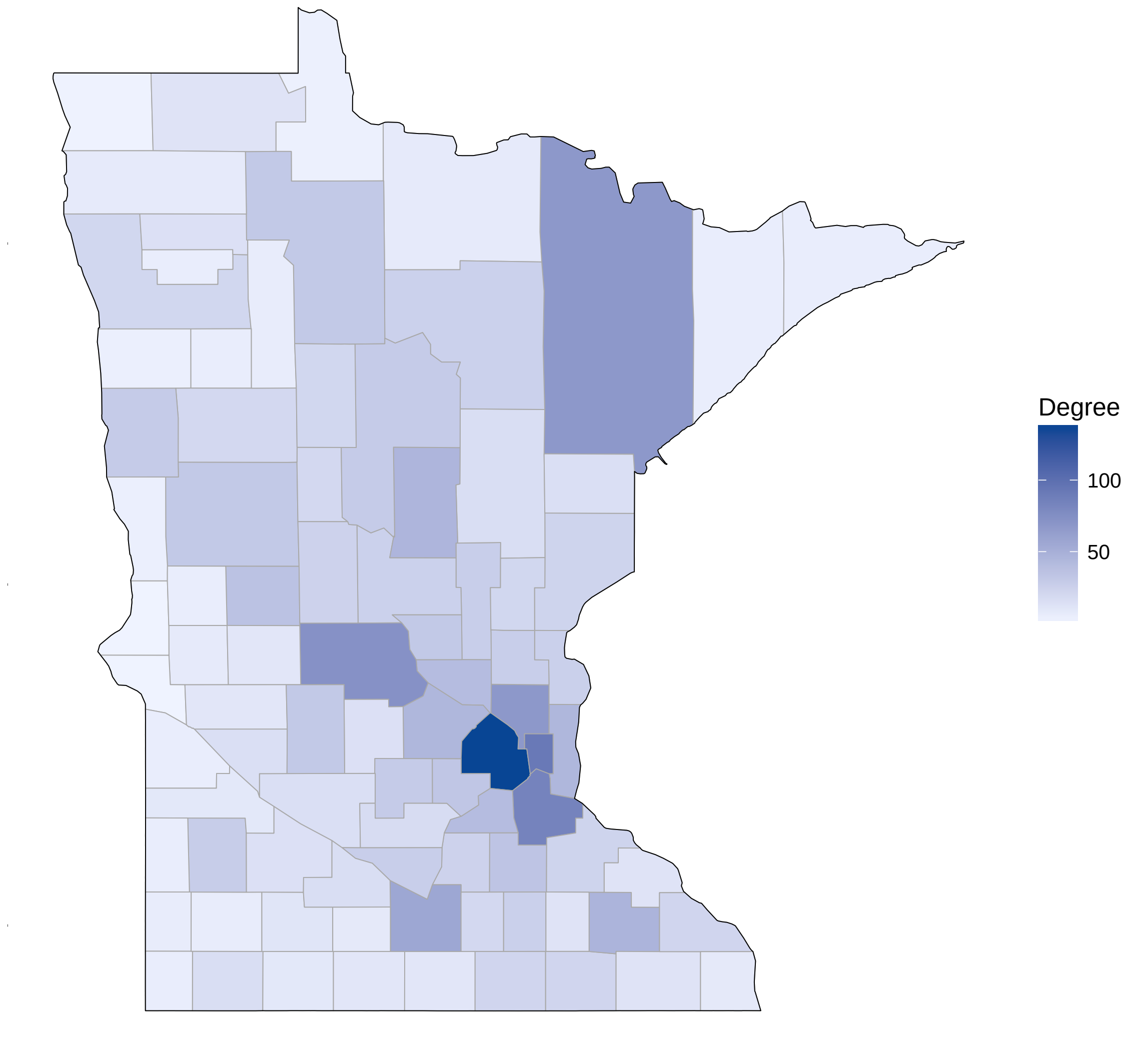


4.19 2010-2011

```
plot(g_1011, layout=layout.fruchterman.reingold(g_1011),
  vertex.size=igraph::degree(g_1011)/7, vertex.label=NA,
  edge.arrow.size=.1, vertex.color="cyan", edge.width=map(c(mn_1011), c(1,5)))
```



```
v2=paste("27", v1, sep="")
d_1011<-igraph::degree(g_1011)
df=data.frame(region=as.numeric(v2), value=as.numeric(d_1011))
county_choropleth(df, state_zoom="minnesota", legend="Degree", num_colors=1)
```



5 Discussion

You'd think I was showing you the same network plots and maps! The migration pattern structure appears to be pretty constant over time. In just about every year, there are one or two counties that aren't connected to any other counties. The counties that had the largest node sizes were Ramsey, Hennepin, and St. Louis counties. These are, perhaps not surprisingly, the counties with the largest cities and populations in the state. It is also worth noting the loops that appeared on the network plots. This indicates that there are people moving around within their own county. This is perhaps most clearly visible on pages 2 and 3, where the diagonal of the adjacency table represents the number of people moving around within that county.

Given all of the societal changes and everything else that happened in the past 20 years, the migration network still is more or less constant. The structure of the network, and the counties with the most nodes, remains the same. The counties with the highest populations and largest metropolitan areas are connected to just about everything else. A lot of social network analysis will yield a network that appears to be clustered into several different groups. For instance, in analyzing the social network of a high school, one might find that the nodes are clustered by grade or social circle, with far more connections within these clusters than between clusters. Interestingly, this isn't the case at the county migration level. The whole state seems to form one big cluster. Some counties are at the center, but there isn't a clear division based on geography or other characteristic.

There are two main considerations here. First, the network does not show an edge if the number of people migrating between two counties is less than 10. The IRS does not report this for privacy reasons, so while there isn't really anything we can do, it is worth keeping in mind. The other main concern is that this network does not show those who moved outside of the state, or even outside of the country. The reason for this is that if we were to take into account the network compromising the entire state, that would grow very large, very quickly. Visualizing the network of 87 nodes can get messy enough; trying to visualize over 3000 nodes would only get harder.

To summarize, there are three main conclusions. First, the migration network patterns between counties tend to generally be the same across time. Secondly, most of the people moving in the state tended to move within their own county, as opposed to moving toward a different county. Lastly, people moving between counties did not seem to be phased by the geographic proximity (or lack thereof) between their counties.

References

[1] Butts C (2015). network: Classes for Relational Data. The Statnet Project `http://statnet.org`. R package version 1.13.0, `http://CRAN.R-project.org/package=network`

[2] Butts C (2008). network: a Package for Managing Relational Data in R. Journal of Statistical Software, 24(2). `http://www.jstatsoft.org/v24/i02/paper`.

[3] Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. `http://igraph.org`.

[4] Hahsler, Michael. Examples for the igraph Package. 2014. `http://michael.hahsler.net/SMU/LearnROnYourOwn/code/igraph.html`