

Statistical Analysis for Story Telling

Recall what we have seen at Feature Extraction:

```
setwd("D:/STOR893-Zhang/vessel-network/feature_extraction/feature")

deg.dat.p2 = read_excel("p2-fro_degreedata.xlsx")
deg.dat.p3 = read_excel("p3-fro_degreedata.xlsx")
deg.dat.p4 = read_excel("p4-fro_degreedata.xlsx")
deg.dat.p5 = read_excel("p5-fro_degreedata.xlsx")
deg.dat.p6 = read_excel("p6-fro_degreedata.xlsx")
deg.dat.p7 = read_excel("p7-fro_degreedata.xlsx")

all.dat.p2 = read_excel("p2-fro_alldata.xlsx")
all.dat.p3 = read_excel("p3-fro_alldata.xlsx")
all.dat.p4 = read_excel("p4-fro_alldata.xlsx")
all.dat.p5 = read_excel("p5-fro_alldata.xlsx")
all.dat.p6 = read_excel("p6-fro_alldata.xlsx")
all.dat.p7 = read_excel("p7-fro_alldata.xlsx")
```

Let's construct node spatial densities by following function.

```
deg_density = function(x,i){
  degree = x$degree
  deg_density = table(degree)[max(degree):1] %>%
    prop.table() %>%
    cumsum() %>%
    rev()
  dt.frame = data.frame(miceage = rep(paste0("P",i,sep=""),max(degree)),
    degree = 1:max(degree),
    density = deg_density)

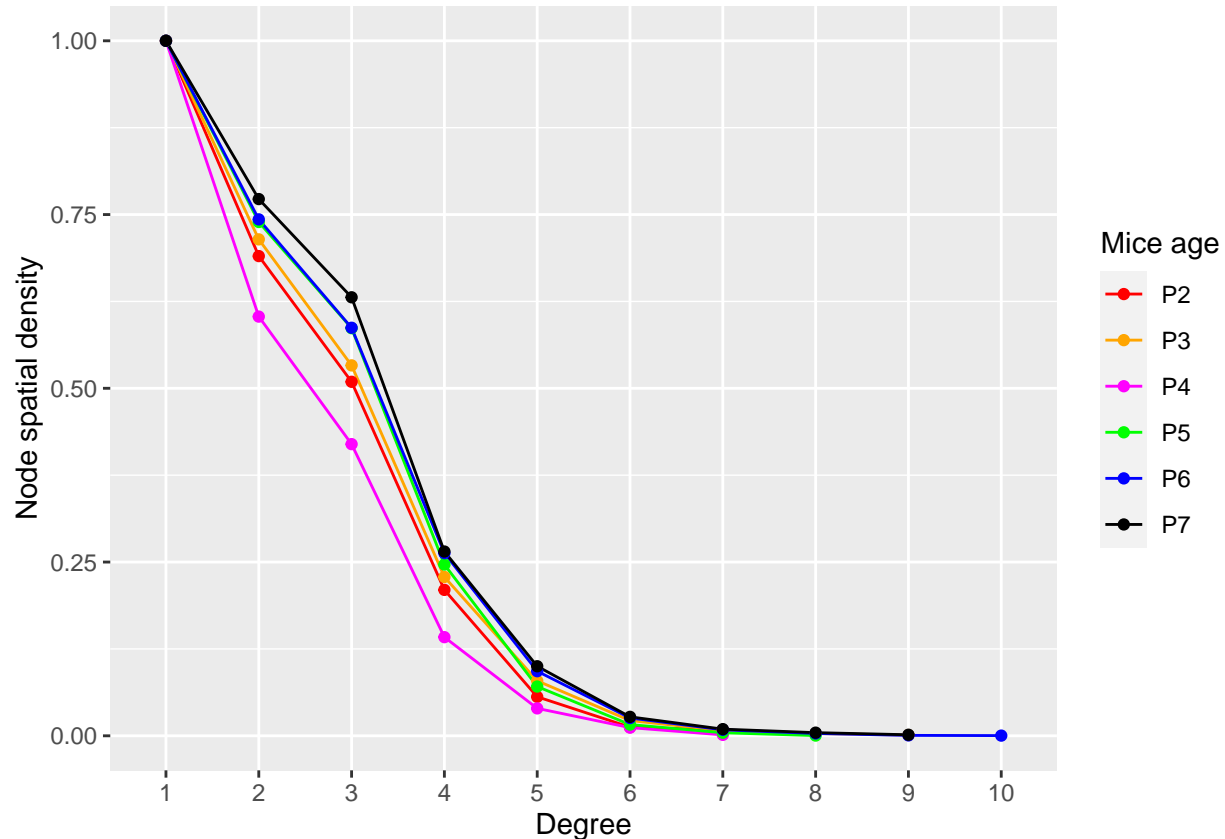
  return(dt.frame)
}

deg.density.p2 = deg_density(deg.dat.p2,2)
deg.density.p3 = deg_density(deg.dat.p3,3)
deg.density.p4 = deg_density(deg.dat.p4,4)
deg.density.p5 = deg_density(deg.dat.p5,5)
deg.density.p6 = deg_density(deg.dat.p6,6)
deg.density.p7 = deg_density(deg.dat.p7,7)
deg_spatial_density = rbind(deg.density.p2,
  deg.density.p3,
  deg.density.p4,
  deg.density.p5,
  deg.density.p6,
  deg.density.p7)
```

Plot node spatial densities. y -axis stands for the density of branching points exceeds degree x , which is the number k of vessels branching out from the single node.

```
deg_spatial_density$miceage = as.factor(deg_spatial_density$miceage)
deg_spatial_density$degree = as.factor(deg_spatial_density$degree)
G.spatial = ggplot(deg_spatial_density, aes(x = degree, y = density, color = miceage, group = miceage))
  geom_line() +
  geom_point() +
  labs(x = "Degree",
       y = "Node spatial density") +
  scale_color_manual(name="Mice age", values=c("red", "orange", "magenta", "green", "blue", "black"))

G.spatial
```



The question is why the degree of spatial density at P4 is lower than the others. We can make a conjecture that at postnatal day 4, there is something happen at the retina of the small mouse.

First, define the function to construct adjacency matrices of P2 through P7. Here is the function for Constructing adjacency matrix:

```
Adj_mat_generate = function(x){

  data.tmp = x

  identified_nodes_key = sort(unique(rbind(data.tmp$node1,data.tmp$node2)[2,]),decreasing=TRUE)
  node1 = sort(x$node1,decreasing=TRUE)
  node2 = sort(x$node2,decreasing=TRUE)

  d = length(identified_nodes_key)
  Adj_mat = matrix(NA,nrow=d,ncol=d)
```

```

rownames(Adj_mat) = identified_nodes_key
colnames(Adj_mat) = identified_nodes_key

for (i in 1:d){
  j = 1;
  while ( j <= i ){
    if (i == j){
      Adj_mat[i,j] = 0
    }
    else{

      key.from = identified_nodes_key[i]
      key.to = identified_nodes_key[j]

      Adj_mat[i,j] = length(node1[which(node1 == key.from & node2 == key.to)])
      Adj_mat[j,i] = Adj_mat[i,j]

    }
    j = j+1
  }
  # Ticker: can be commentized
  # print(i)
}

return(list(Adj_mat))
}

```

By using the function above, we can construct adjacency matrices.

```

p2.clean = Adj_mat_generate(all.dat.p2)
p3.clean = Adj_mat_generate(all.dat.p3)
p4.clean = Adj_mat_generate(all.dat.p4)
p5.clean = Adj_mat_generate(all.dat.p5)
p6.clean = Adj_mat_generate(all.dat.p6)
p7.clean = Adj_mat_generate(all.dat.p7)

```

Next, Construct the graph structure based on adjacency matrices by

```

G2 = graph_from_adjacency_matrix(p2.clean[[1]],mode="undirected")
G3 = graph_from_adjacency_matrix(p3.clean[[1]],mode="undirected")
G4 = graph_from_adjacency_matrix(p4.clean[[1]],mode="undirected")
G5 = graph_from_adjacency_matrix(p5.clean[[1]],mode="undirected")
G6 = graph_from_adjacency_matrix(p6.clean[[1]],mode="undirected")
G7 = graph_from_adjacency_matrix(p7.clean[[1]],mode="undirected")

```

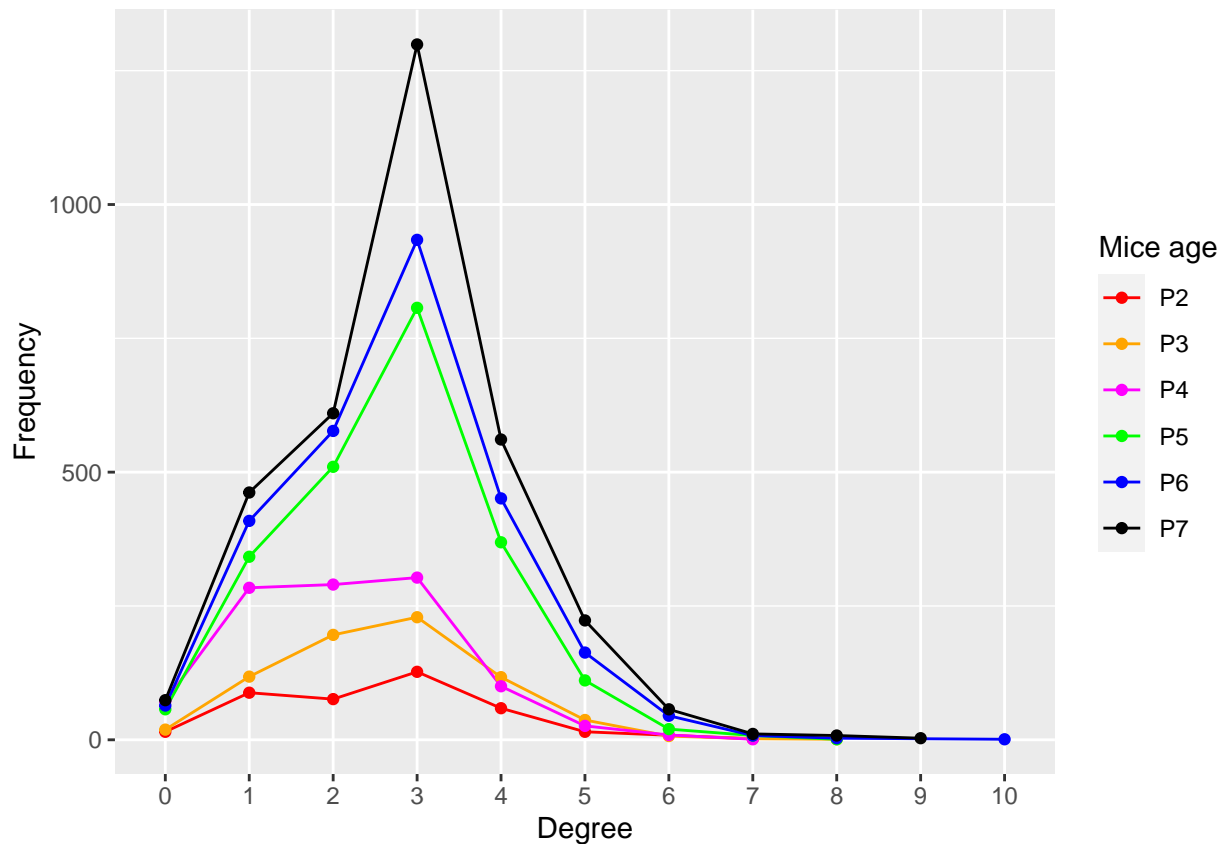
Note that the plot below has the same information but is drawn in different way. Plot degree densities are following.

```

dt.degree.p2 = data.frame(as.data.frame(table(degree(G2))), P=rep("P2",length(table(degree(G2)))))
dt.degree.p3 = data.frame(as.data.frame(table(degree(G3))), P=rep("P3",length(table(degree(G3)))))
dt.degree.p4 = data.frame(as.data.frame(table(degree(G4))), P=rep("P4",length(table(degree(G4)))))
dt.degree.p5 = data.frame(as.data.frame(table(degree(G5))), P=rep("P5",length(table(degree(G5)))))
dt.degree.p6 = data.frame(as.data.frame(table(degree(G6))), P=rep("P6",length(table(degree(G6)))))
dt.degree.p7 = data.frame(as.data.frame(table(degree(G7))), P=rep("P7",length(table(degree(G7)))))
dt.degree = rbind(dt.degree.p2,dt.degree.p3,dt.degree.p4,dt.degree.p5,dt.degree.p6,dt.degree.p7)
names(dt.degree) = c("Deg","Freq","P")

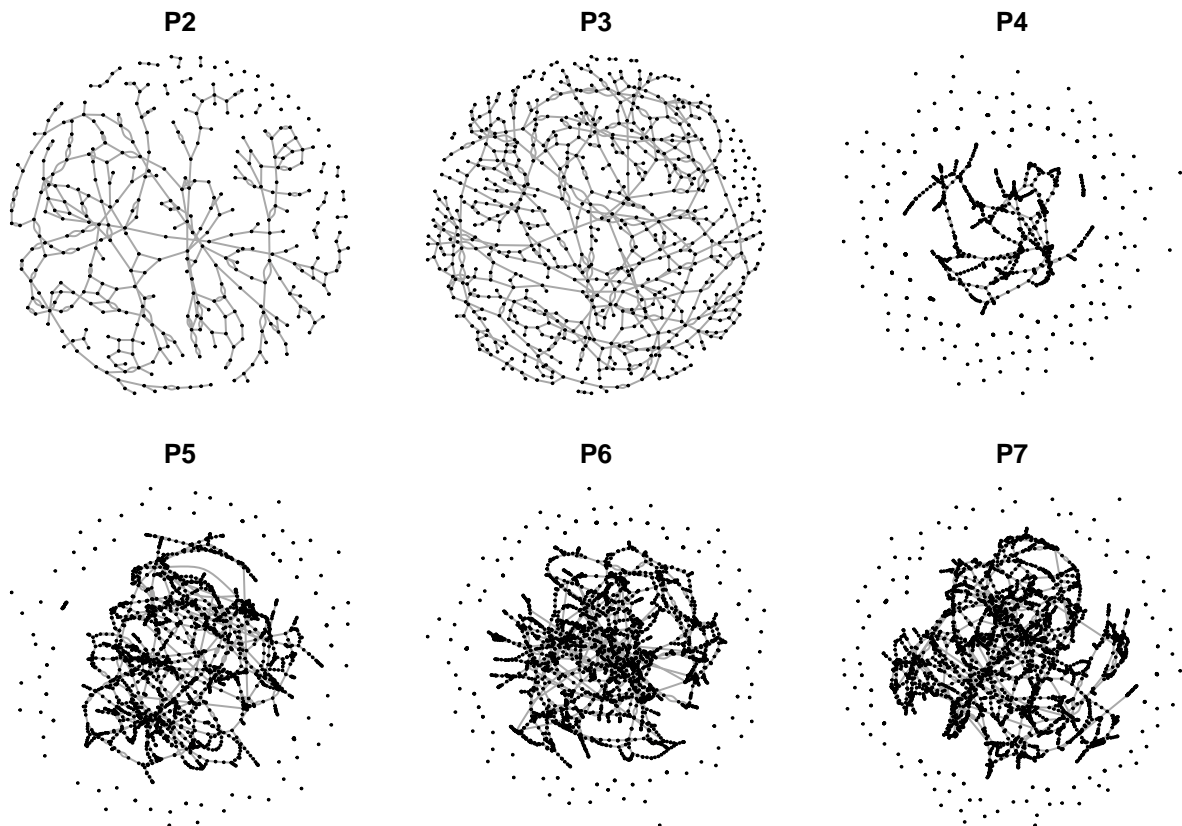
```

```
G.deg = ggplot(dt.degree, aes(y=Freq,x=factor(Deg),color=factor(P),group=factor(P))) +
  ylab("Frequency") +
  scale_x_discrete(name = "Degree", limits=c("0","1","2","3","4","5","6","7","8","9","10")) +
  scale_color_manual(name="Mice age",values=c("red","orange","magenta","green","blue","black")) +
  geom_line()+
  geom_point()
G.deg
```



Here are the graph plots before clustering. From P4, we can see the shape of graph does change a lot. Next, plot graphical representation of networks:

```
par(mar=c(1,1,1,1))
par(mfrow=c(2,3))
plot(G2,vertex.label=NA,vertex.size=1,vertex.color="red",main="P2")
plot(G3,vertex.label=NA,vertex.size=1,vertex.color="orange",main="P3")
plot(G4,vertex.label=NA,vertex.size=1,vertex.color="magenta",main="P4")
plot(G5,vertex.label=NA,vertex.size=1,vertex.color="green",main="P5")
plot(G6,vertex.label=NA,vertex.size=1,vertex.color="blue",main="P6")
plot(G7,vertex.label=NA,vertex.size=1,vertex.color="black",main="P7")
```



We run clustering by using edge betweenness. As we can see by checking the number of groups clustered by edge betweenness, there are explosively many groups appearing at P4.

```
Cl.betw.2 = cluster_edge_betweenness(graph=G2)
Cl.betw.3 = cluster_edge_betweenness(graph=G3)
Cl.betw.4 = cluster_edge_betweenness(graph=G4)
Cl.betw.5 = cluster_edge_betweenness(graph=G5)
Cl.betw.6 = cluster_edge_betweenness(graph=G6)
Cl.betw.7 = cluster_edge_betweenness(graph=G7)

length(sort(unique(Cl.betw.2$membership),decreasing=FALSE))

## [1] 49

length(sort(unique(Cl.betw.3$membership),decreasing=FALSE))

## [1] 54

length(sort(unique(Cl.betw.4$membership),decreasing=FALSE))

## [1] 151

length(sort(unique(Cl.betw.5$membership),decreasing=FALSE))

## [1] 110

length(sort(unique(Cl.betw.6$membership),decreasing=FALSE))

## [1] 127
```

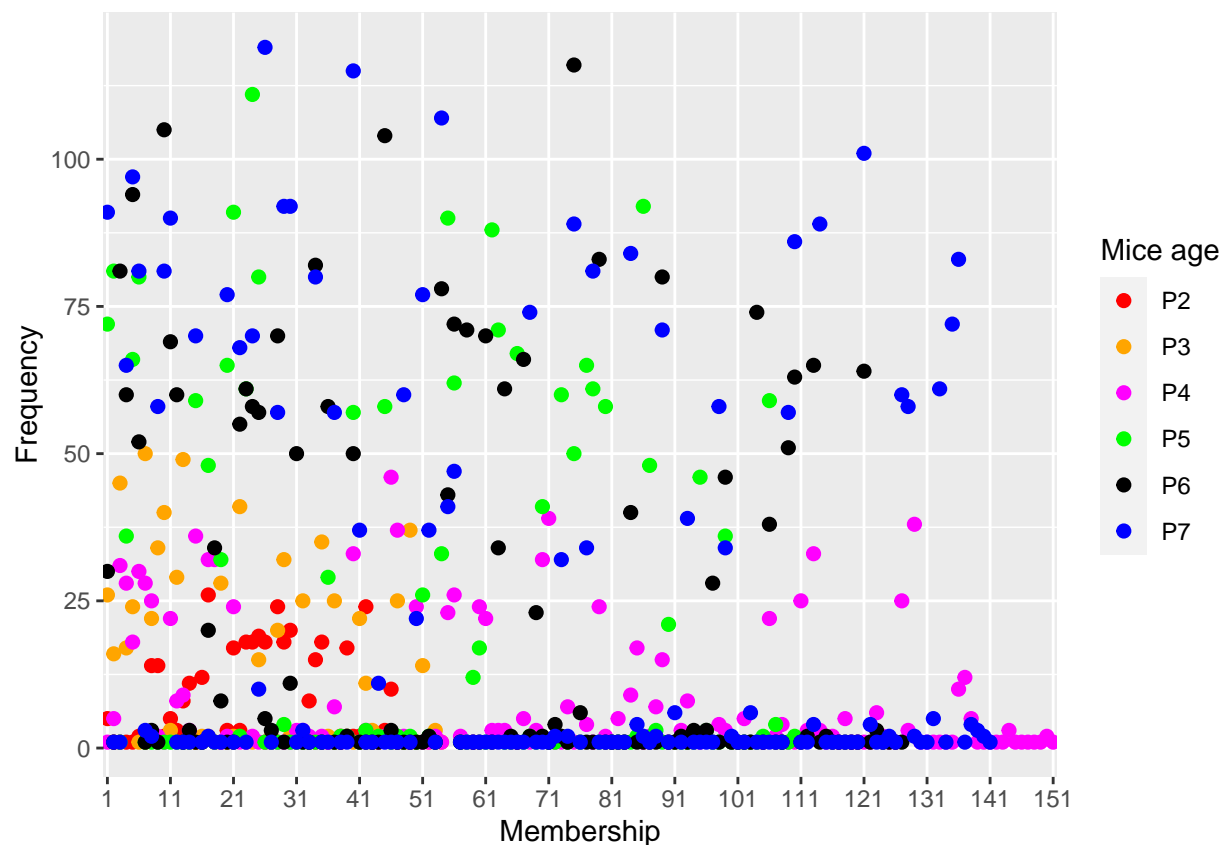
```
length(sort(unique(Cl.betw.7$membership),decreasing=FALSE))
```

```
## [1] 141
```

We can check that for each P , how many members are joining into a single membership. We can observe that the membership status are similar between $P2, P3$ and $P4$ while $P5, P6$, and $P7$ have similarity by themselves but show difference from the previous ones.

```
dt.membership.p2 = data.frame(as.data.frame(table(Cl.betw.2$membership)),
                              P=rep("P2",length(table(Cl.betw.2$membership))))
dt.membership.p3 = data.frame(as.data.frame(table(Cl.betw.3$membership)),
                              P=rep("P3",length(table(Cl.betw.3$membership))))
dt.membership.p4 = data.frame(as.data.frame(table(Cl.betw.4$membership)),
                              P=rep("P4",length(table(Cl.betw.4$membership))))
dt.membership.p5 = data.frame(as.data.frame(table(Cl.betw.5$membership)),
                              P=rep("P5",length(table(Cl.betw.5$membership))))
dt.membership.p6 = data.frame(as.data.frame(table(Cl.betw.6$membership)),
                              P=rep("P6",length(table(Cl.betw.6$membership))))
dt.membership.p7 = data.frame(as.data.frame(table(Cl.betw.7$membership)),
                              P=rep("P7",length(table(Cl.betw.7$membership))))
dt.membership = rbind(dt.membership.p2,
                      dt.membership.p3,
                      dt.membership.p4,
                      dt.membership.p5,
                      dt.membership.p6,
                      dt.membership.p7)
names(dt.membership) = c("Membership", "Freq", "P")

G.membership = ggplot(dt.membership, aes(y=Freq,x=factor(Membership),color=factor(P))) +
  scale_x_discrete(breaks=seq(1,190,10))+
  ylab("Frequency") +
  xlab("Membership") +
  scale_color_manual(name="Mice age",values=c("red","orange","magenta","green","black","blue")) +
  geom_point(cex=2)
G.membership
```



We do another clustering by louvain to secure the guarantee that all clustering results lead us to the same observation. The below code is for checking the number of groups clustered by louvain:

```
C1.louvain.2 = cluster_louvain(graph=G2)
C1.louvain.3 = cluster_louvain(graph=G3)
C1.louvain.4 = cluster_louvain(graph=G4)
C1.louvain.5 = cluster_louvain(graph=G5)
C1.louvain.6 = cluster_louvain(graph=G6)
C1.louvain.7 = cluster_louvain(graph=G7)
```

Number of groups clustered by louvain:

```
length(sort(unique(C1.louvain.2$membership),decreasing=FALSE))
```

```
## [1] 48
```

```
length(sort(unique(C1.louvain.3$membership),decreasing=FALSE))
```

```
## [1] 56
```

```
length(sort(unique(C1.louvain.4$membership),decreasing=FALSE))
```

```
## [1] 148
```

```
length(sort(unique(C1.louvain.5$membership),decreasing=FALSE))
```

```
## [1] 117
```

```
length(sort(unique(C1.louvain.6$membership),decreasing=FALSE))
```

```
## [1] 130
```

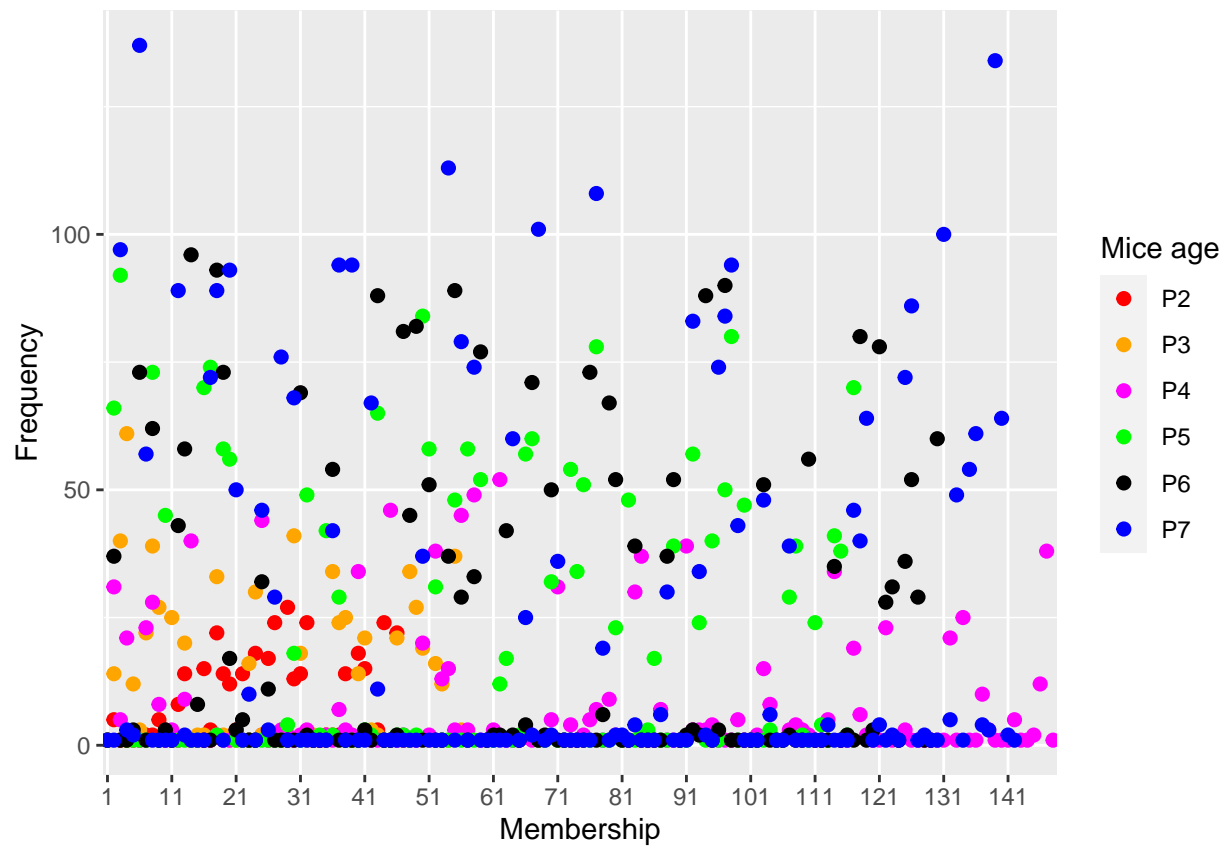
```
length(sort(unique(Cl.louvain.7$membership),decreasing=FALSE))
```

```
## [1] 142
```

Finally, we do plotting membership distributions from the louvain clustering.

```
dt.membership.louvain.p2 = data.frame(as.data.frame(table(Cl.louvain.2$membership)),
                                       P=rep("P2",length(table(Cl.louvain.2$membership))))
dt.membership.louvain.p3 = data.frame(as.data.frame(table(Cl.louvain.3$membership)),
                                       P=rep("P3",length(table(Cl.louvain.3$membership))))
dt.membership.louvain.p4 = data.frame(as.data.frame(table(Cl.louvain.4$membership)),
                                       P=rep("P4",length(table(Cl.louvain.4$membership))))
dt.membership.louvain.p5 = data.frame(as.data.frame(table(Cl.louvain.5$membership)),
                                       P=rep("P5",length(table(Cl.louvain.5$membership))))
dt.membership.louvain.p6 = data.frame(as.data.frame(table(Cl.louvain.6$membership)),
                                       P=rep("P6",length(table(Cl.louvain.6$membership))))
dt.membership.louvain.p7 = data.frame(as.data.frame(table(Cl.louvain.7$membership)),
                                       P=rep("P7",length(table(Cl.louvain.7$membership))))
dt.membership.louvain = rbind(dt.membership.louvain.p2,
                              dt.membership.louvain.p3,
                              dt.membership.louvain.p4,
                              dt.membership.louvain.p5,
                              dt.membership.louvain.p6,
                              dt.membership.louvain.p7)
names(dt.membership.louvain) = c("Membership","Freq","P")

G.membership.louvain = ggplot(dt.membership.louvain, aes(y=Freq,x=factor(Membership),color=factor(P))) +
  scale_x_discrete(breaks=seq(1,190,10))+
  ylab("Frequency") +
  xlab("Membership") +
  scale_color_manual(name="Mice age",values=c("red","orange","magenta","green","black","blue")) +
  geom_point(cex=2)
G.membership.louvain
```

The two results show the same consequences. Hence, we can visually find the “change point”.