Load necessary packages

```
source('GraphBuilder.R')
library(reshape2)
library(ggplot2)
library(NetworkDistance)
library(sna)
library(gridExtra)
library(docstring)
theme_set(theme_classic(base_size = 10))
```

Define function for later analyses

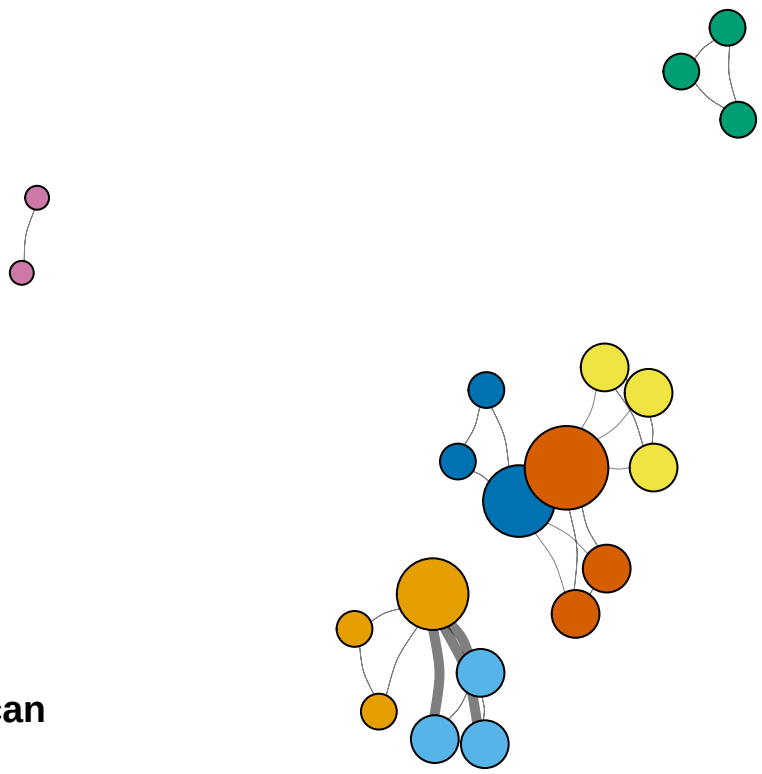
```
density.calc <- function(g){

  d <- edge_density(g) # calculate unweighted density of graph
  dw <- edge_density(g) * sum(E(g)$weight) / length(E(g))

  return(c(d, dw))
}

vars.calc <- function(g){
  n.members <- table(V(g)$group)
  groups <- names(n.members[n.members > 2])
  g.sub <- induced_subgraph(g, which(V(g)$group %in% groups))
  g.sub <- subgraph.edges(g.sub, eids = which(E(g.sub)$group == 'within'))

  within.var <- as_long_data_frame(g.sub) %>%
    group_by(from_group) %>%
    summarize(mean = mean(count),
              sum.squares = sum((count - mean)^2)) %>%
    ungroup() %>%
    summarize(within.var = sum(sum.squares)/(length(V(g.sub)) - length(groups))) %>%
    pull()

  return(within.var)
}

df.create <- function(list.of.graphs, session = NA, lab = NA, week = NA,
                      method, func = 'density'){
  if(func == 'density'){
      df <- as.data.frame(t(matrix(unlist(lapply(list.of.graphs, density.calc)),
                                   ncol = length(list.of.graphs))))
      colnames(df) <- c('density', 'density.weighted')
  } else {
      df <- as.data.frame(t(matrix(unlist(lapply(list.of.graphs, vars.calc)),
                                   ncol = length(list.of.graphs))))
      colnames(df) <- c('within.variance')
  }
```

```
  args = c('session', 'lab', 'week', 'method')
  i = 1
  for(var in list(session, lab, week, method)){
    if(!is.na(var)){
      df[, args[i]] <- var
    }
    i = i + 1
  }

  return(df)
}

reliability <- function(g1, g2, type, normalize = FALSE){
  hd <- nd.hamming(list(as_adjacency_matrix(g1), as_adjacency_matrix(g2)))$`D`[1]
  hd.w <- nd.hamming(list(as_adjacency_matrix(g1, attr = 'weight'),
                          as_adjacency_matrix(g2, attr = 'weight')))$D[1]

  if(normalize){
    max.matrix <- pmax(as_adjacency_matrix(g1, attr = 'weight'),
                       as_adjacency_matrix(g2, attr = 'weight'))

    d <- density.calc(graph_from_adjacency_matrix(max.matrix, mode = type,
                                                  weighted = TRUE))

    hd <- 1 - hd/d[1]
    hd.w <- 1 - hd.w/d[2]
  }
  return(c(hd, hd.w))
}
```

Load scan method graphs

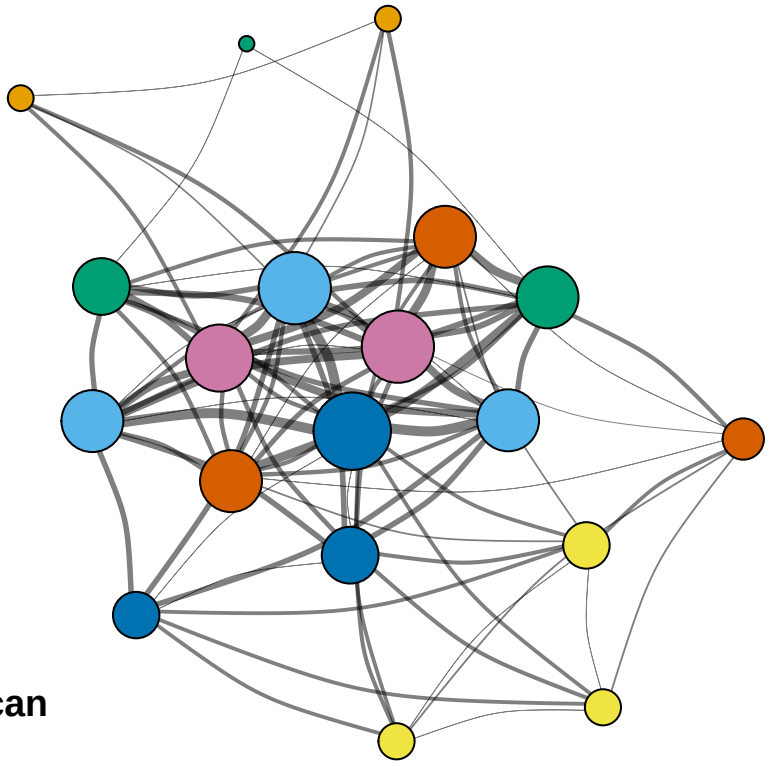
Wednesday section

```
g.scan.9.11 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'scan-student', name = '9-11 Scan')

plot.graph(g.scan.9.11, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```



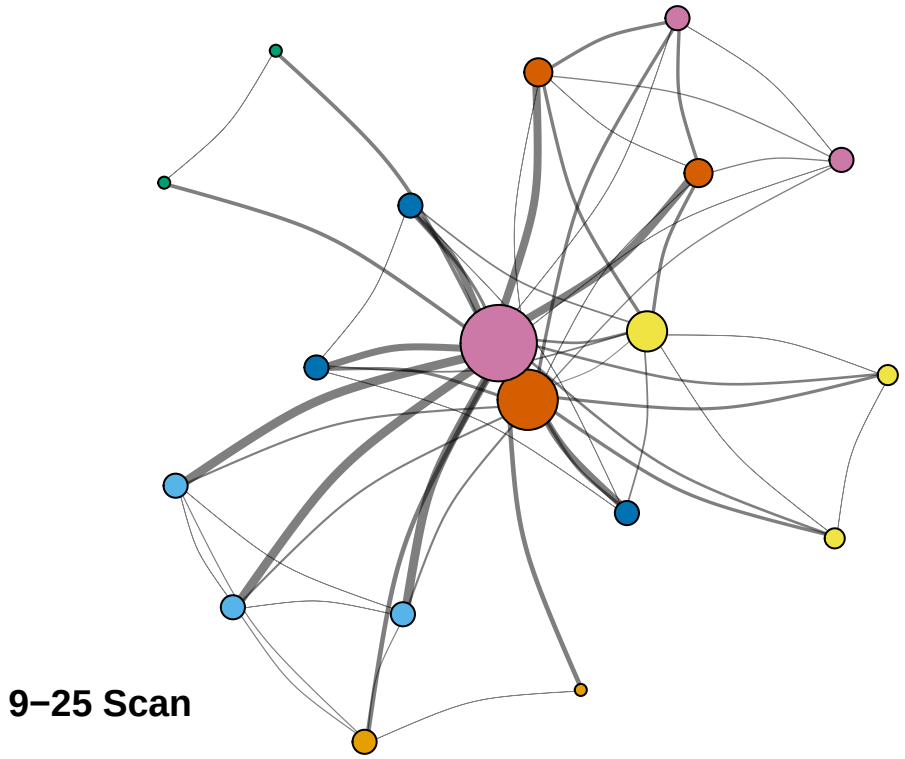
9−11 Scan

```
g.scan.9.18 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'scan-student', name = '9-18 Scan')
plot.graph(g.scan.9.18, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```

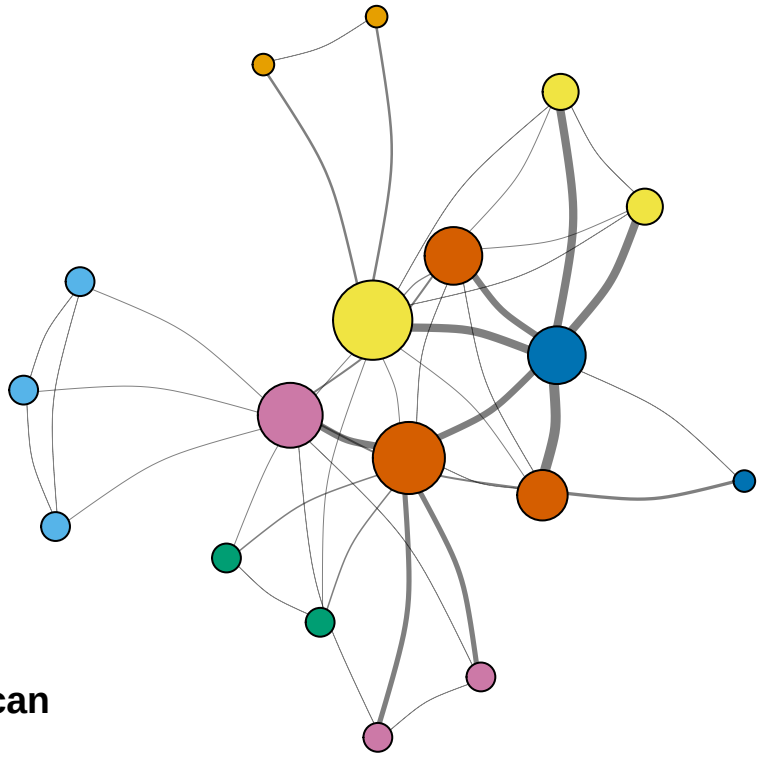


9−18 Scan

```
g.scan.9.25 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'scan-student', name = '9-25 Scan')
plot.graph(g.scan.9.25, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```

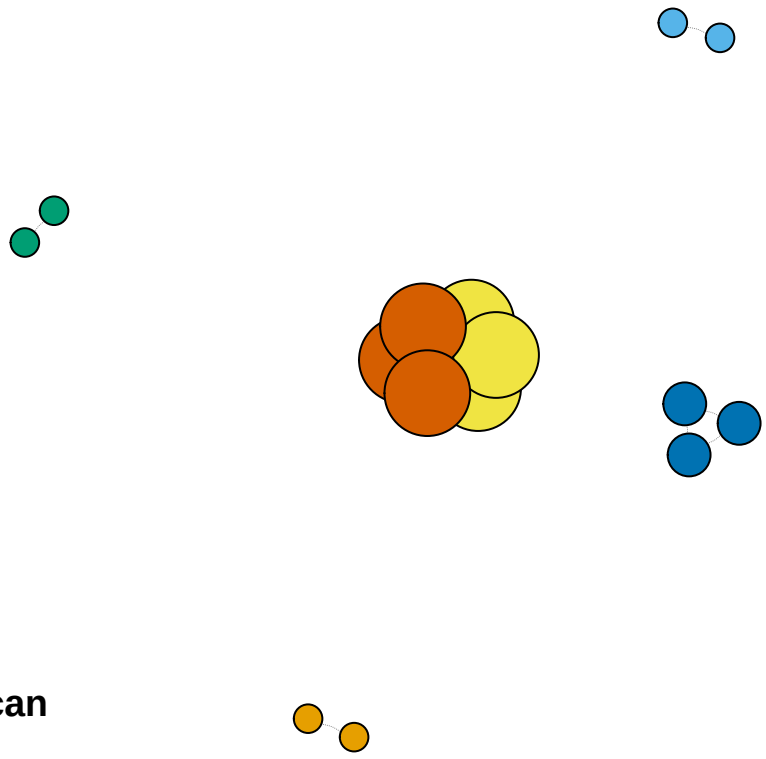



```
g.scan.10.9 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'scan-student', name = '10-8 Scan')
plot.graph(g.scan.10.9, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```



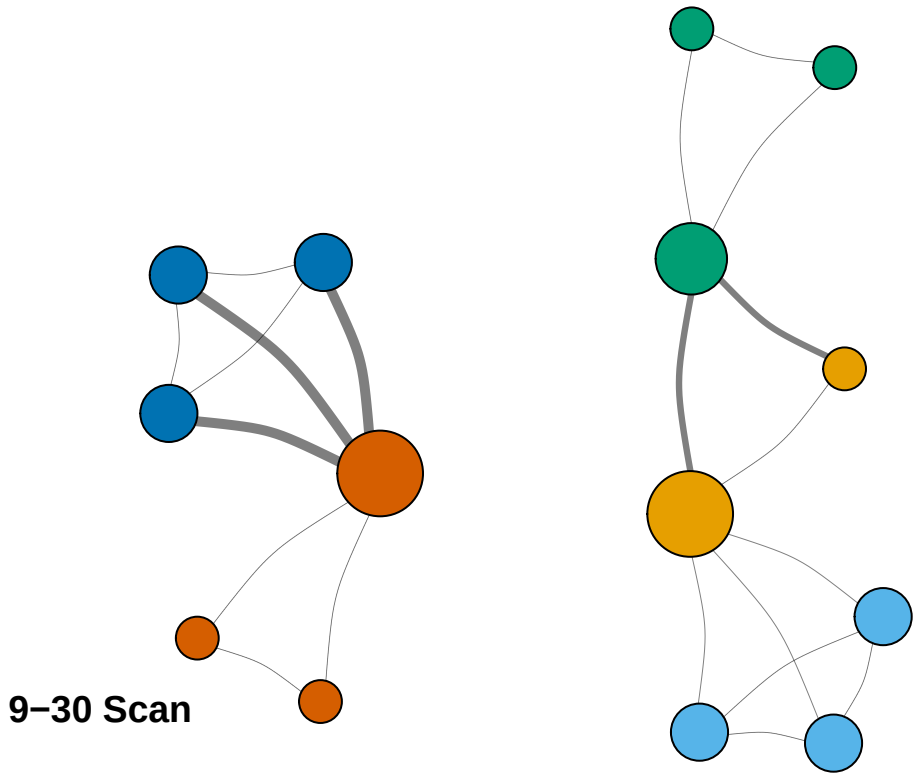
Monday section

```
g.scan.9.16 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'scan-student', name = '9-16 Scan')
plot.graph(g.scan.9.16, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```

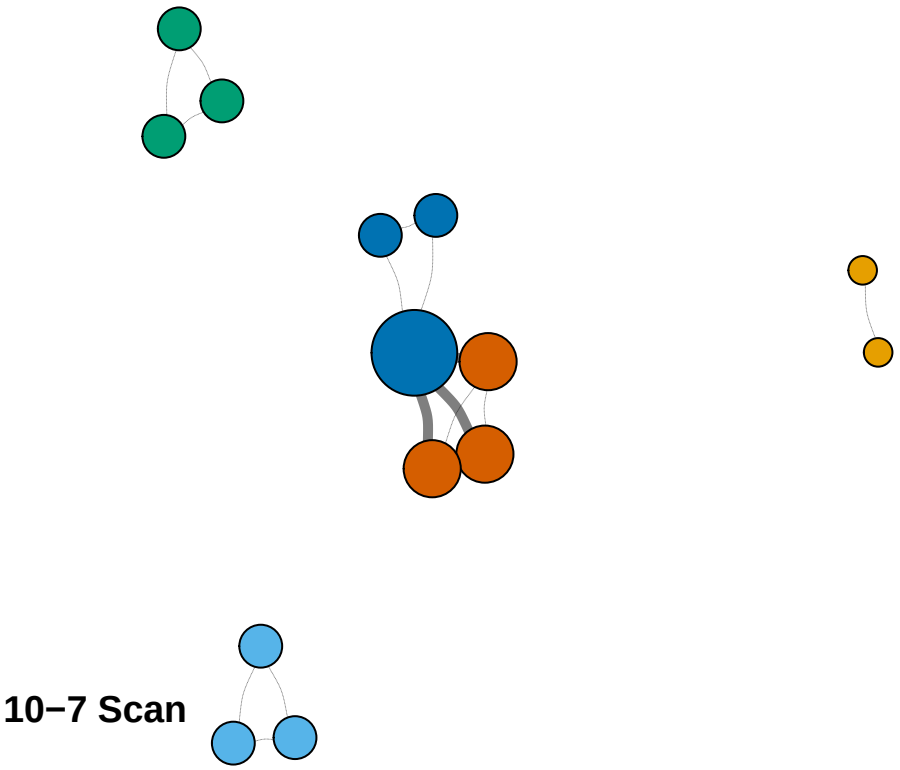


9−16 Scan

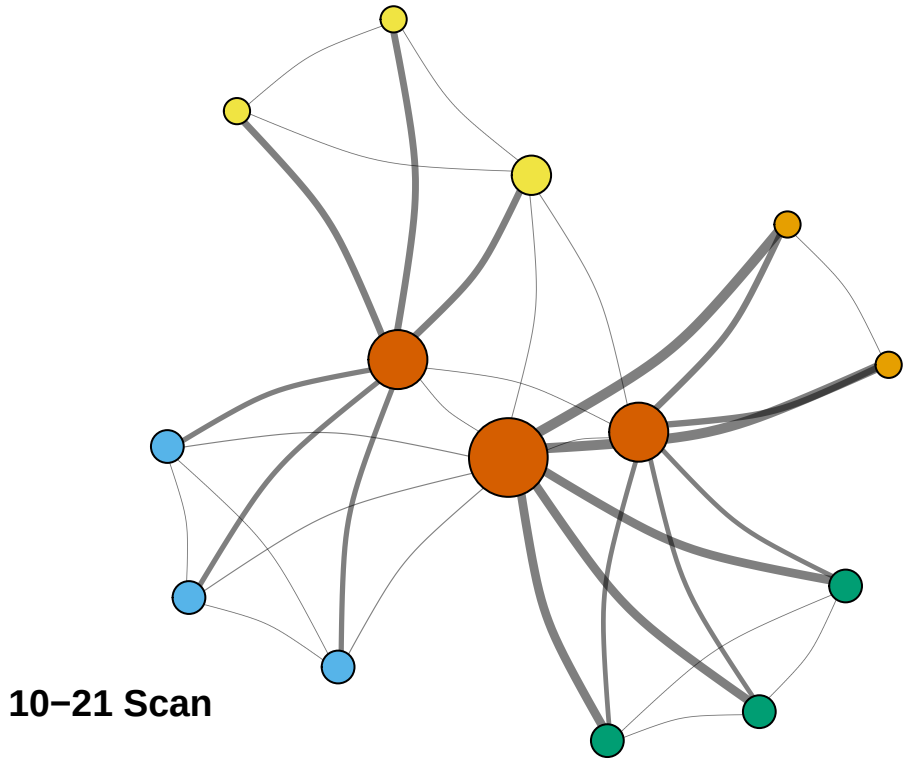

```
g.scan.9.30 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'scan-student', name = '9-30 Scan')
plot.graph(g.scan.9.30, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```




```
g.scan.10.7 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'scan-student', name = '10-7 Scan')
plot.graph(g.scan.10.7, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```



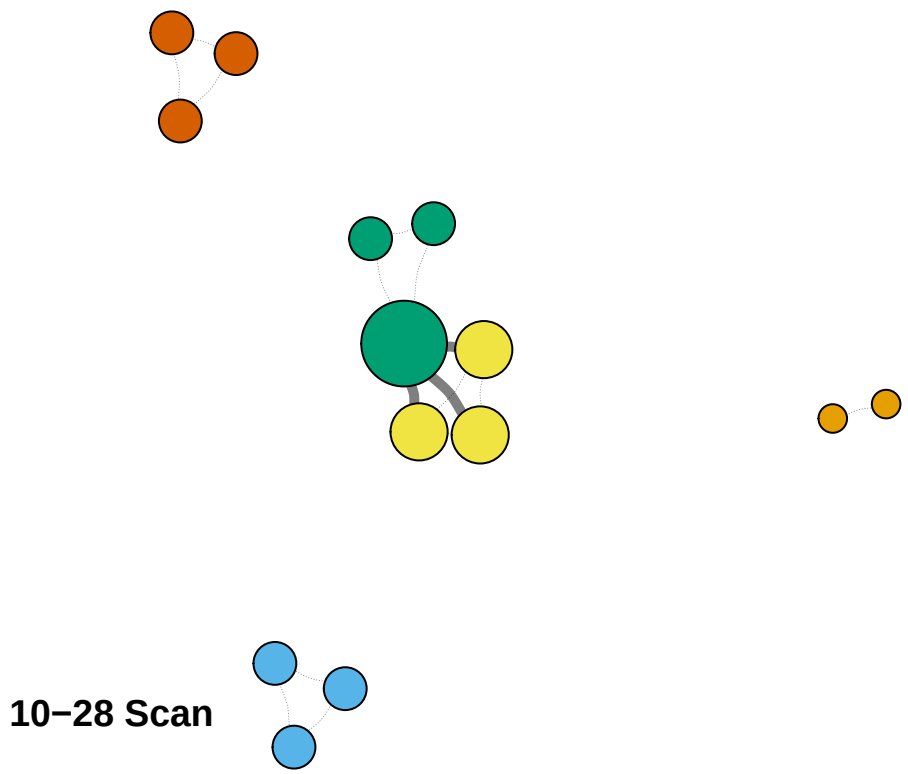

```
g.scan.10.21 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_
                                     method = 'scan-student', name = '10-21 Scan')
plot.graph(g.scan.10.21, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```




```
g.scan.10.28 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_
                                     method = 'scan-student', name = '10-28 Scan')
# extra everyone node coded, so we'll remove that
g.scan.10.28 <- subgraph(g.scan.10.28, 1:(length(V(g.scan.10.28)) - 1))
```

```
## Warning in subgraph(g.scan.10.28, 1:(length(V(g.scan.10.28)) - 1)): At
## structural_properties.c:1984 :igraph_subgraph is deprecated from igraph
## 0.6, use igraph_induced_subgraph instead
```

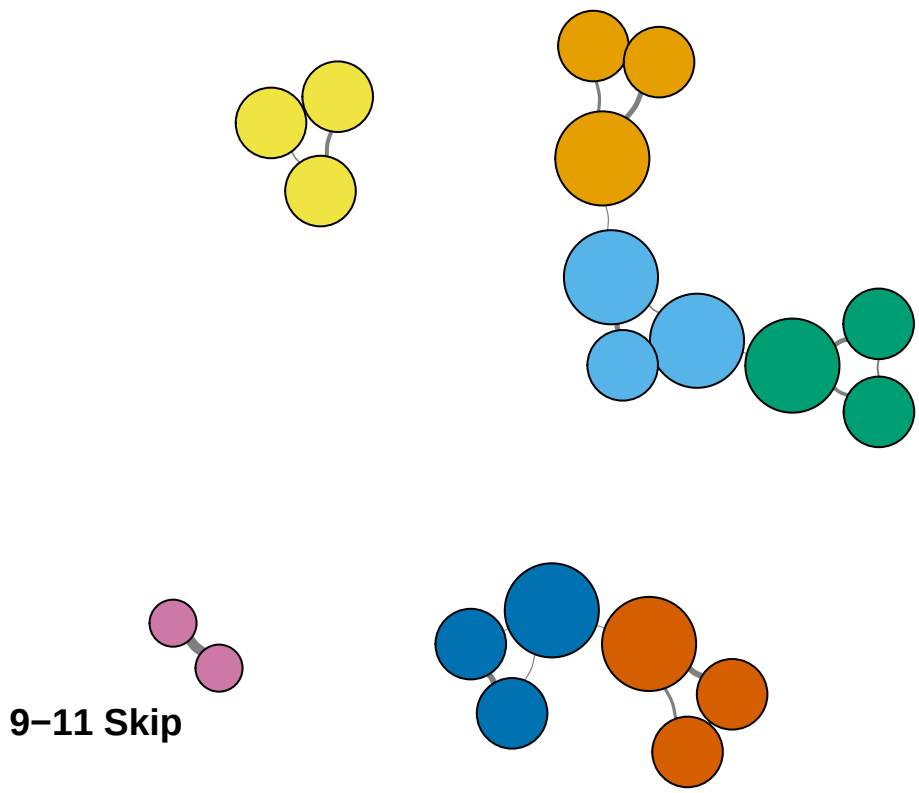
```
plot.graph(g.scan.10.28, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```



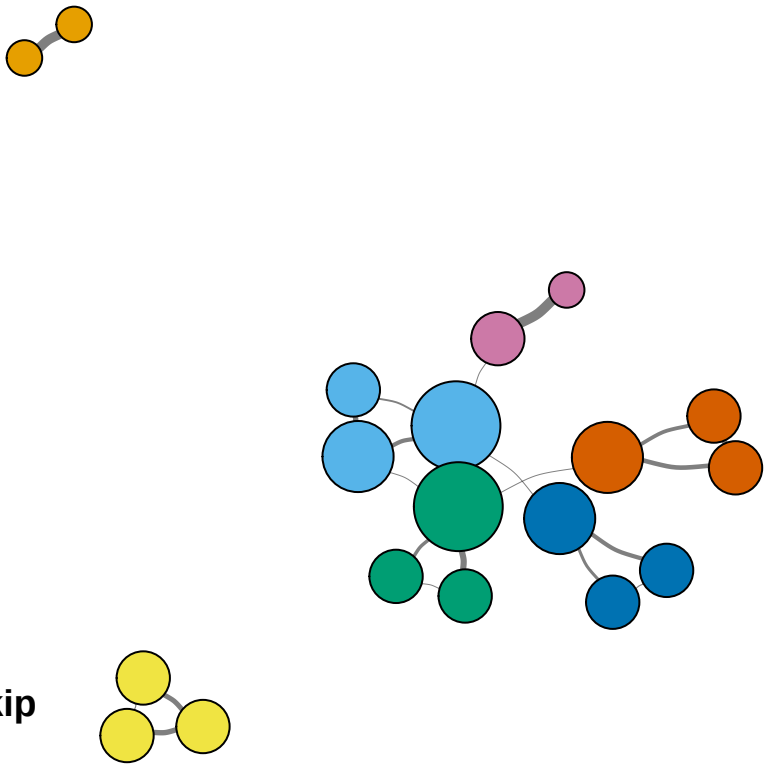
Load skip method graphs

Wednesday section

```
g.skip.9.11 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_F
                                    method = 'skip', name = '9-11 Skip')
plot.graph(g.skip.9.11, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```

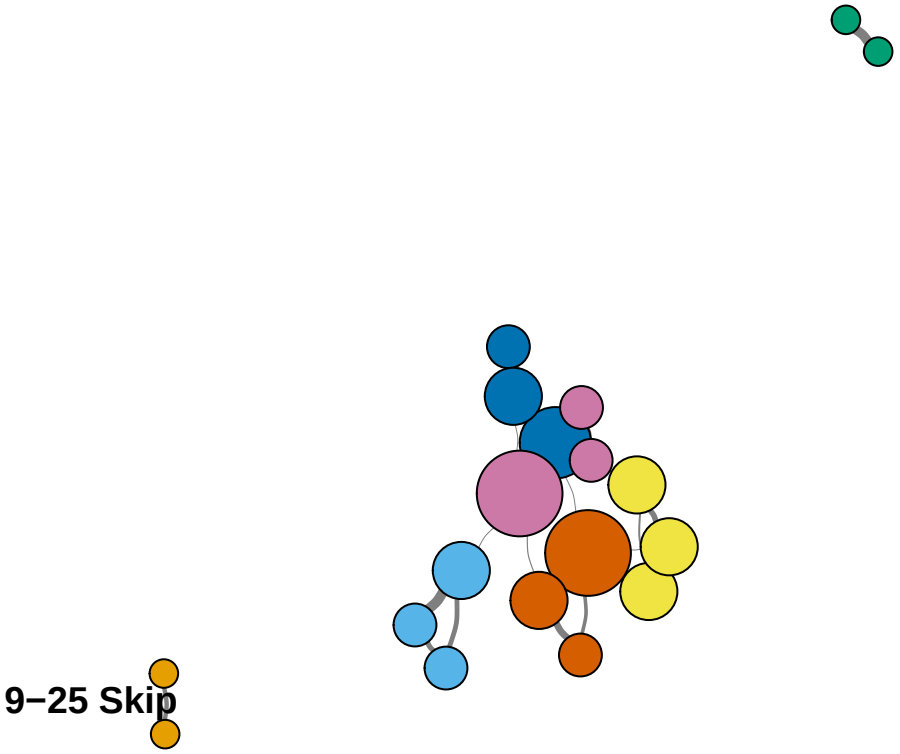



```
g.skip.9.18 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'skip', name = '9-18 Skip')

# change names to line up between raters when computing reliability
V(g.skip.9.18)$name <- c('1A', '1B', '2C', '2B', '2A', '3A', '3C', '3B', '4A',
                         '4B', '4C', '5A', '5B', '5C', '6A', '6B', '6C', '7A',
                         '7B')

plot.graph(g.skip.9.18, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```

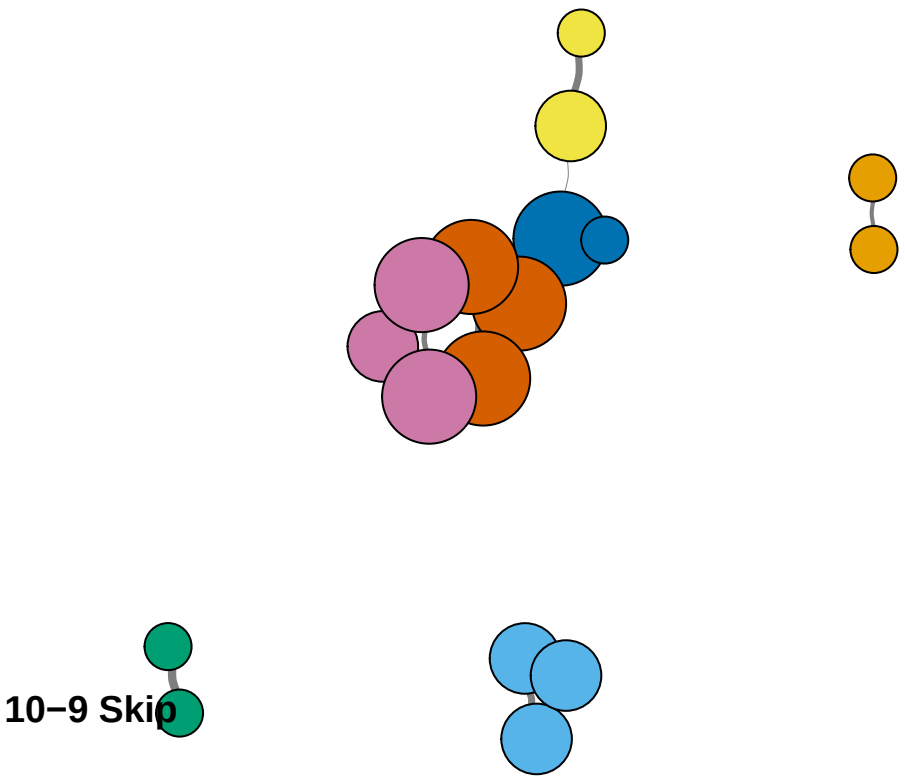


9−18 Skip


```
g.skip.9.25 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'skip', name = '9-25 Skip')
plot.graph(g.skip.9.25, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```




```
g.skip.10.9 <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1112_Fa
                                    method = 'skip', name = '10-9 Skip')
plot.graph(g.skip.10.9, vertex.scale = 20, edge.scale = 5, vertex.label = FALSE)
```



Check interrater reliability for 9-18

```
g.scan.9.18.DK <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P111
                                       method = 'scan-student')
g.scan.9.18.MS <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P111
                                       method = 'scan-student')
V(g.scan.9.18.MS)$name <- c('1A', '1B', '2A', '2B', '2C', '3A', '3B', '3C', '4A', '4B', '4C', '5A', '5B

print('Reliability of scan method between CW and DK for 9-18...')
```

```
## [1] "Reliability of scan method between CW and DK for 9-18..."
```

```
reliability(g.scan.9.18, g.scan.9.18.DK, type = 'undirected')
```

```
## [1] 0.03508772 4.21052632
```

```
reliability(g.scan.9.18, g.scan.9.18.DK, type = 'undirected', normalize = TRUE)
```

```
## [1] 0.9310345 0.8537180
```

```
print('Reliability of scan method between CW and MS for 9-18...')
```

```
## [1] "Reliability of scan method between CW and MS for 9-18..."
```

```
reliability(g.scan.9.18, igraph::permute(g.scan.9.18.MS,
                                         match(V(g.scan.9.18.MS)$name,
                                               V(g.scan.9.18)$name)),
            type = 'undirected')
```

```
## [1] 0.06432749 3.66666667
```

```
reliability(g.scan.9.18, igraph::permute(g.scan.9.18.MS,
                                         match(V(g.scan.9.18.MS)$name,
                                               V(g.scan.9.18)$name)),
            type = 'undirected', normalize = TRUE)
```

```
## [1] 0.8842105 0.8809343
```

```
print('Reliability of scan method between DK and MS for 9-18...')
```

```
## [1] "Reliability of scan method between DK and MS for 9-18..."
```

```
reliability(g.scan.9.18.DK, igraph::permute(g.scan.9.18.MS,
                                         match(V(g.scan.9.18.MS)$name,
                                               V(g.scan.9.18.DK)$name)),
            type = 'undirected')
```

```
## [1] 0.0877193 6.0877193
```

```
reliability(g.scan.9.18.DK, igraph::permute(g.scan.9.18.MS,
                                         match(V(g.scan.9.18.MS)$name,
                                               V(g.scan.9.18.DK)$name)),
            type = 'undirected', normalize = TRUE)
```

```
## [1] 0.8421053 0.8010321
```

```
g.skip.9.18.CW <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P111
                                       method = 'skip')

print('Reliability of skip method between CW and DK for 9-18...')
```

```
## [1] "Reliability of skip method between CW and DK for 9-18..."
```

```
reliability(g.skip.9.18, igraph::permute(g.skip.9.18.CW,
                                         match(V(g.skip.9.18.CW)$name,
                                               V(g.skip.9.18)$name)),
            type = 'undirected')
```

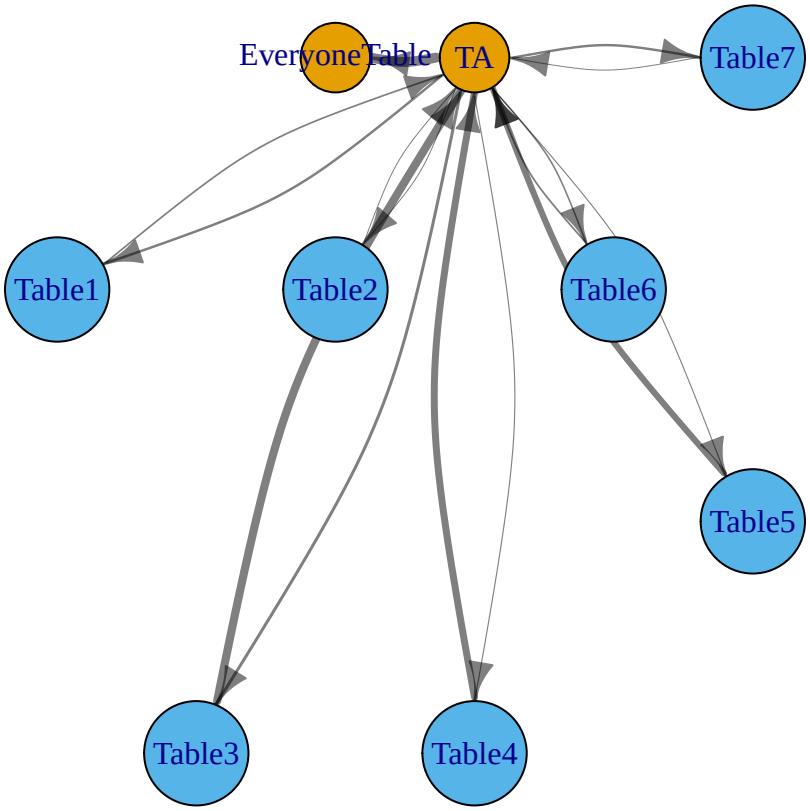
```
## [1] 0.01754386 0.20467836
```

```
reliability(g.skip.9.18, igraph::permute(g.skip.9.18.CW,
                                         match(V(g.skip.9.18.CW)$name,
                                               V(g.skip.9.18)$name)),
            type = 'undirected', normalize = TRUE)
```

```
## [1] 0.8571429 0.8292683
```

Check interrater reliability for TA graphs

```
g.TA.9.18.Dur.DK <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1
                                         method = 'scan-group', directed = TRUE)

g.TA.9.18.Dur.DW <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P1
                                         method = 'scan-group', directed = TRUE)
plot.graph(g.TA.9.18.Dur.DW, vertex.scale = 20, edge.scale = 5, layout = 'B22',
           standardNodes = c('TA', 'EveryoneTable'))
```



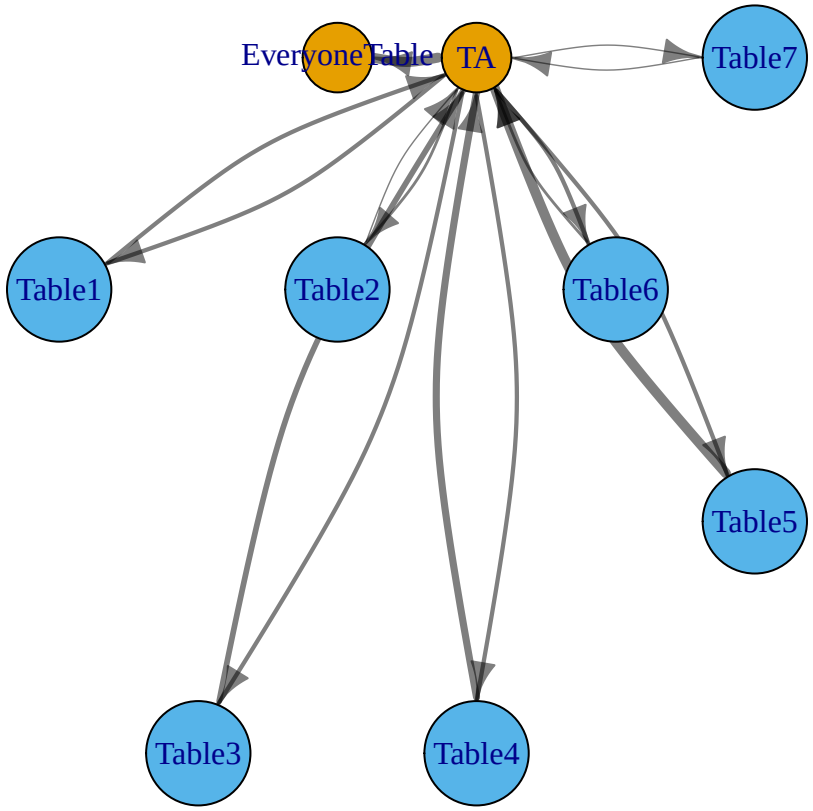

```
reliability(g.TA.9.18.Dur.DK, g.TA.9.18.Dur.DW, type = 'directed')
```

```
## [1] 0.000000 1.056361
```

```
reliability(g.TA.9.18.Dur.DK, g.TA.9.18.Dur.DW, type = 'directed',
            normalize = TRUE)
```

```
## [1] 1.0000000 0.9892407
```

```
g.TA.9.18.No.DK <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P11
                                        method = 'scan-group', directed = TRUE)

g.TA.9.18.No.DW <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P11
                                         method = 'scan-group', directed = TRUE)
plot.graph(g.TA.9.18.No.DW, vertex.scale = 20, edge.scale = 5, layout = 'B22',
           standardNodes = c('TA', 'EveryoneTable'))
```




```
reliability(g.TA.9.18.No.DK, g.TA.9.18.No.DW, type = 'directed')
```

```
## [1] 0.00000000 0.02777778
```

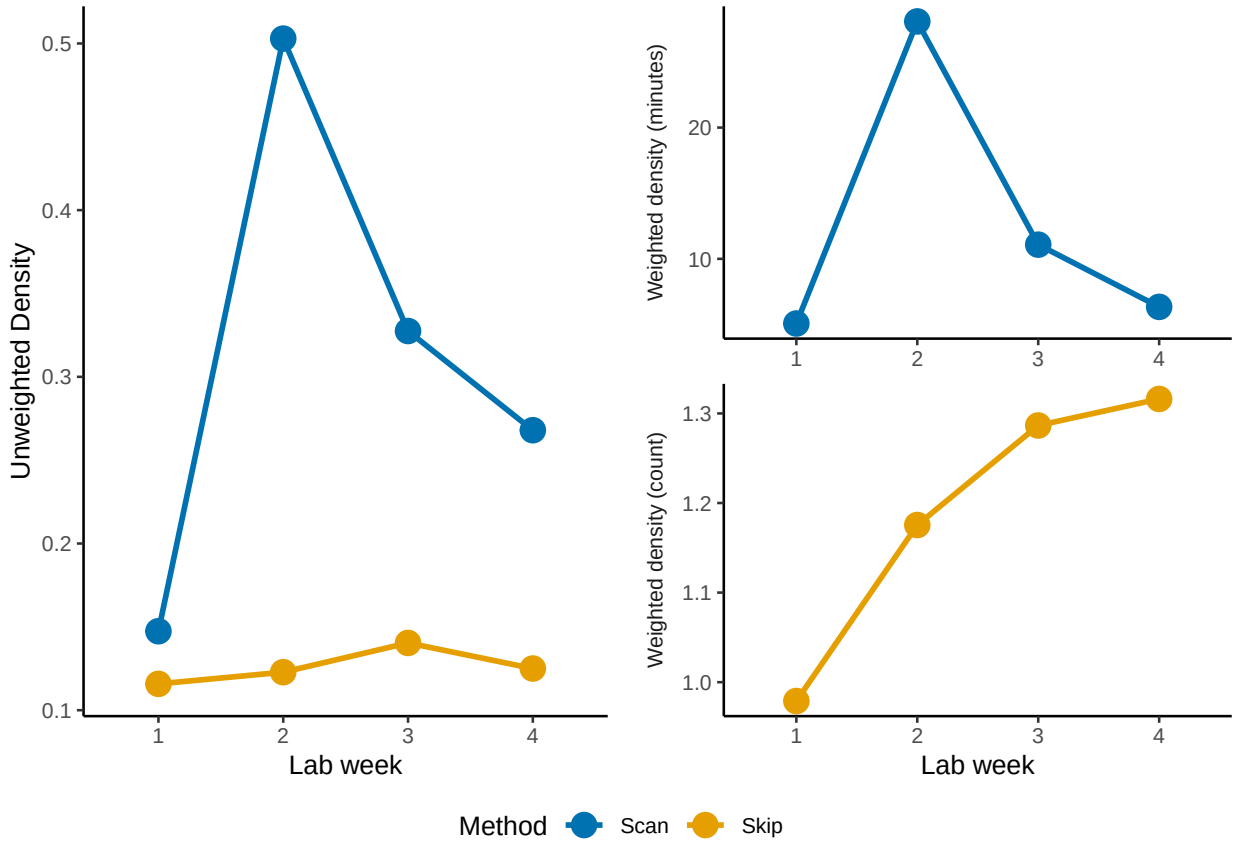
```
reliability(g.TA.9.18.No.DK, g.TA.9.18.No.DW, type = 'directed', normalize = TRUE)
```

```
## [1] 1.0000000 0.9591837
```

Analyze evolution of density over time

```
get_legend<-function(myggplot){
  tmp <- ggplot_gtable(ggplot_build(myggplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)
}

scan.graphs.wednesday <- list(g.scan.9.11, g.scan.9.18, g.scan.9.25, g.scan.10.9)
skip.graphs.wednesday <- list(g.skip.9.11, g.skip.9.18, g.skip.9.25, g.skip.10.9)
scan.graphs.monday <- list(g.scan.9.16, g.scan.9.30, g.scan.10.7, g.scan.10.21,
                           g.scan.10.28)

df.density <- rbind(df.create(scan.graphs.wednesday, session = 'Wednesday',
                              lab = c(1, 1, 2, 2), week = c(1, 2, 1, 2), method = 'scan'),
                    df.create(skip.graphs.wednesday, session = 'Wednesday',
                              lab = c(1, 1, 2, 2), week = c(1, 2, 1, 2), method = 'skip')) %>%
  mutate(lab.week = paste('L', lab, 'W', week, sep = ''),
         method.session = paste('M', method, 'L', session)) %>%
  melt(., measure.vars = c('density', 'density.weighted'))

plot.unweight <- ggplot(df.density %>%
         filter(variable == 'density'),
         aes(x = as.factor(lab.week), y = value, group = method.session,
             color = as.factor(method))) +
  geom_point(size = 4) +
  geom_line(stat = 'identity', size = 1) +
  labs(x = 'Lab week', y = 'Unweighted Density') +
  scale_color_manual(name = 'Method', values = c('#0072B2', '#E69F00'),
                     labels = c('Scan', 'Skip')) +
  scale_shape_manual(name = 'Lab Number', values = c(16, 15)) +
  theme(legend.position = "none") +
  scale_x_discrete(labels = c(1, 2, 3, 4))

plot.weight <- ggplot(df.density %>%
                        filter(variable == 'density.weighted') %>%
                        mutate(method = ifelse(method == 'scan', 'one', 'two')),
                      aes(x = as.factor(lab.week), y = value,
                          group = method.session, color = as.factor(method))) +
  geom_point(size = 4) +
  geom_line(stat = 'identity', size = 1) +
  facet_wrap(~method, scales = 'free', nrow = 2, strip.position = "left",
               labeller = as_labeller(c(one = "Weighted density (minutes)",
                                        two = "Weighted density (count)"))) +
  labs(x = 'Lab week') +
  scale_color_manual(name = 'Method', values = c('#0072B2', '#E69F00'),
                     labels = c('Scan', 'Skip')) +
  scale_shape_manual(name = 'Lab session', values = c(16, 15)) +
  theme(strip.background = element_blank(),
        strip.placement = "outside") +
  ylab(NULL) +
  theme(legend.position = "bottom") +
```

```
  scale_x_discrete(labels = c(1, 2, 3, 4))

legend <- get_legend(plot.weight)
plot.weight <- plot.weight + theme(legend.position = "none")

grid.arrange(plot.unweight, plot.weight, legend, layout_matrix = rbind(c(1, 2), c(3, 3)),
             widths = c(2.7, 2.7), heights = c(2.5, 0.2))
```



ADDITIONAL WORK: Reliability of group-level scan method

Generate adjacency matrices from BORIS files

```
g <- boris.to.adjacency(file1 = 'C:/Users/Cole/Box Sync/Network analysis/Exported_Data/P1116_Fall2017/S
                        nvid1 = 25, method = 'scan-group', directed = TRUE,
                        filename = 'Directed/Scan_Pendulum1_401_MS_adj.csv')

g <- boris.to.adjacency(file1 = 'C:/Users/Cole/Box Sync/Network analysis/Exported_Data/P1116_Fall2017/S
                        nvid1 = 25, method = 'scan-group', directed = TRUE,
                        filename = 'Directed/Scan_Pendulum1_401_CW_adj.csv')
```

Read adjacency matrices and calculate reliability

```
g.scan.group.MS <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P11
                                        method = 'scan-group', name = 'MS scan-group')

g.scan.group.CW <- graph.from.adjacency('C:/Users/Cole/Box Sync/Network analysis/Adjacency_Matrices/P11
                                        method = 'scan-group', name = 'CW scan-group')

reliability(g.scan.group.MS, g.scan.group.CW, type = 'directed',
            normalize = FALSE)
```

```
## [1]  0.01818182 34.38501818
```

```
reliability(g.scan.group.MS, g.scan.group.CW, type = 'directed',
            normalize = TRUE)
```

```
## [1] 0.9545455 0.7501649
```