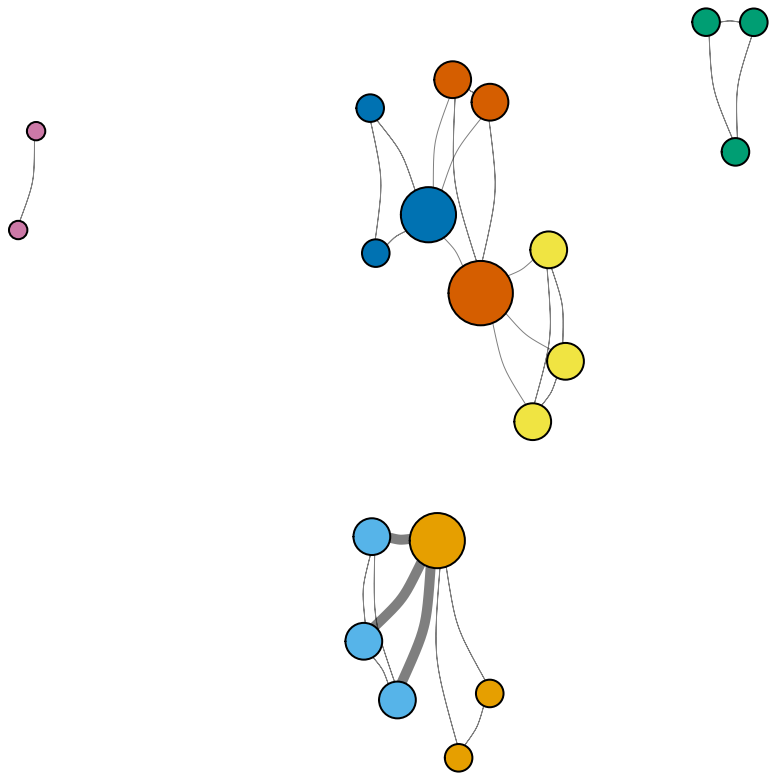
```r
source('GraphBuilder.R')
library(reshape2)
library(ggplot2)
library(NetworkDistance)
library(sna)
library(gridExtra)
theme_set(theme_classic(base_size = 10))
```

```r
density.calc <- function(g, method = 1){
  d <- edge_density(g)
  dw <- ifelse(method == 1, edge_density(g) * sum(E(g)$time) / length(E(g)), edge_density(g) * sum(E(g)$

  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 = 1,
                      func = 'density'){
  if(func == 'density'){
      df <- as.data.frame(t(matrix(unlist(lapply(list.of.graphs, density.calc, method = method)),
                                   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
  }
```

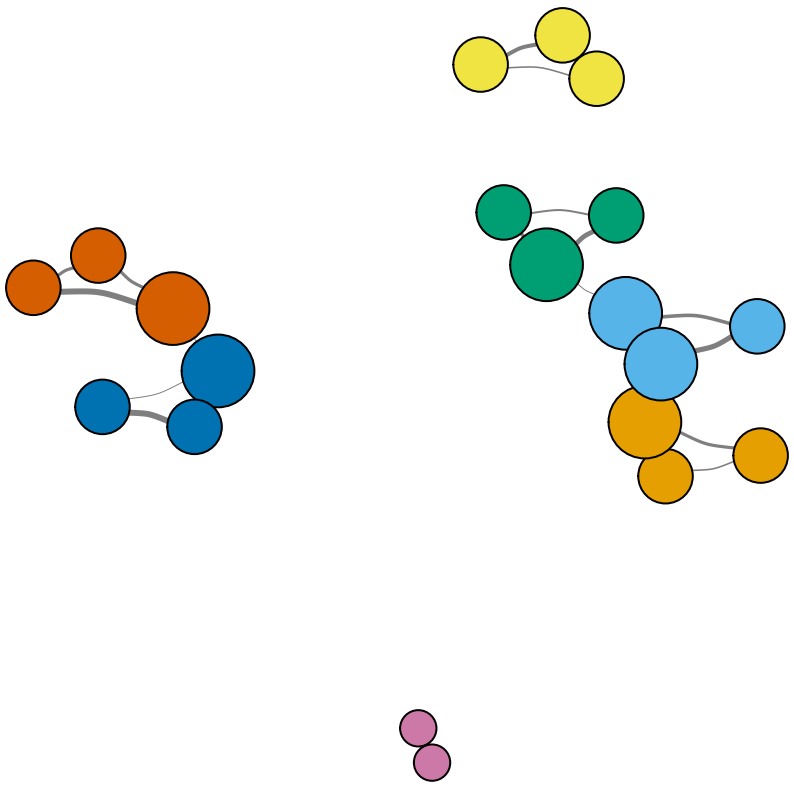
```
  return(df)
}

reliability <- function(g1, g2, method = 1, normalize = FALSE){
  hd <- nd.hamming(list(as_adjacency_matrix(g1), as_adjacency_matrix(g2)))$`D`[1]
  if(method == 1){
    weight = 'time'
  } else {
    weight = 'count'
  }
  hd.w <- nd.hamming(list(as_adjacency_matrix(g1, attr = weight),
                          as_adjacency_matrix(g2, attr = weight)))$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 = 'undirected', weighted = weight), r
    hd <- 1 - hd/d[1]
    hd.w <- 1 - hd.w/d[2]
  }
  return(c(hd, hd.w))
}
```

9-11

```
g.inter.9.11 <- create.graph(file1 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/V
                             file2 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/V
                             nvid2 = 6)
plot.graph(g.inter.9.11)
```

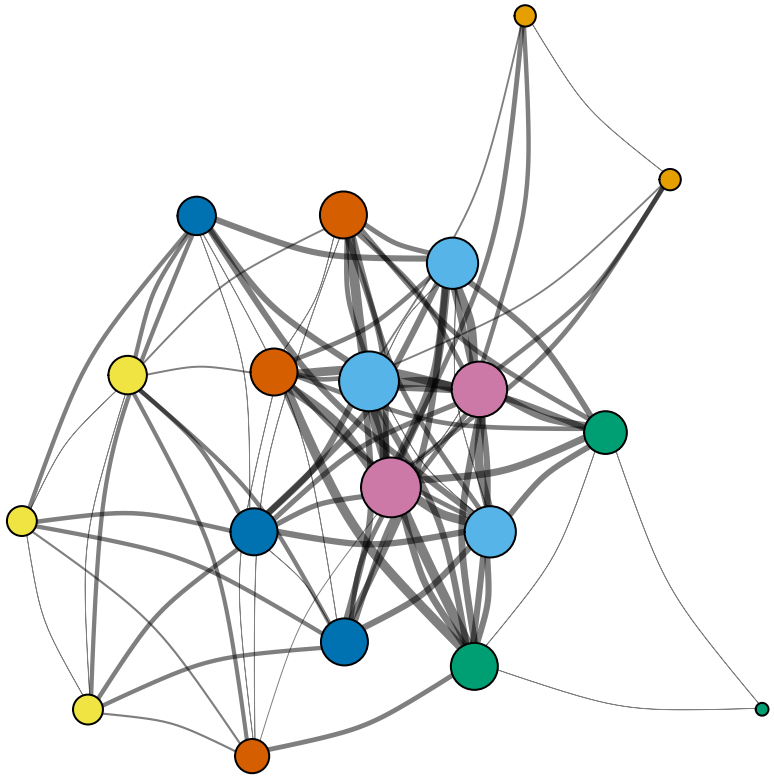


```
g.intra.9.11 <- graph.from.adjacency('C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Da
plot.graph(g.intra.9.11)
```

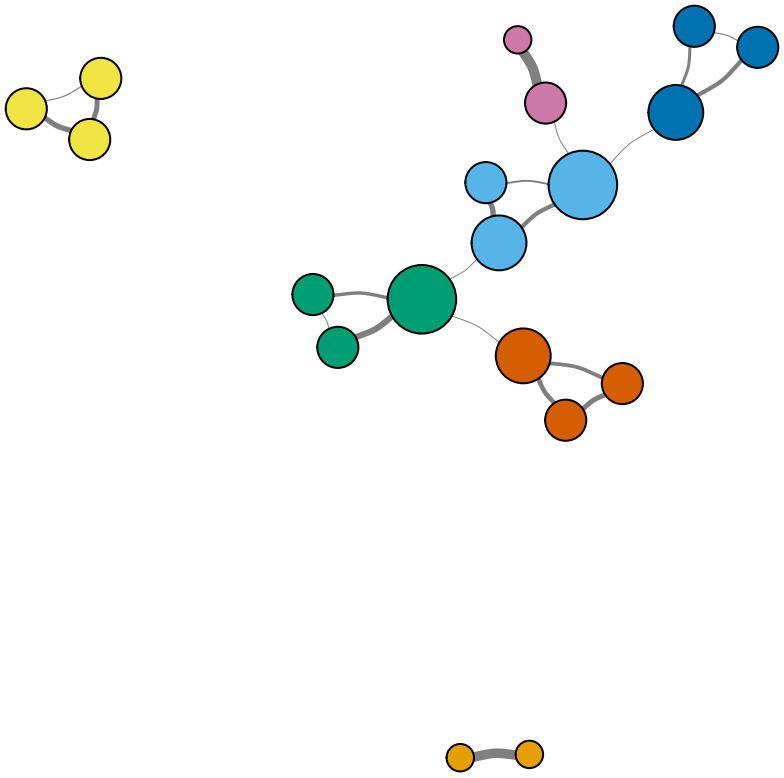


9-18

```
g.inter.9.18 = create.graph(file1 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Vi
                            file2 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Vi
                            nvid2 = 7)
plot.graph(g.inter.9.18)
```

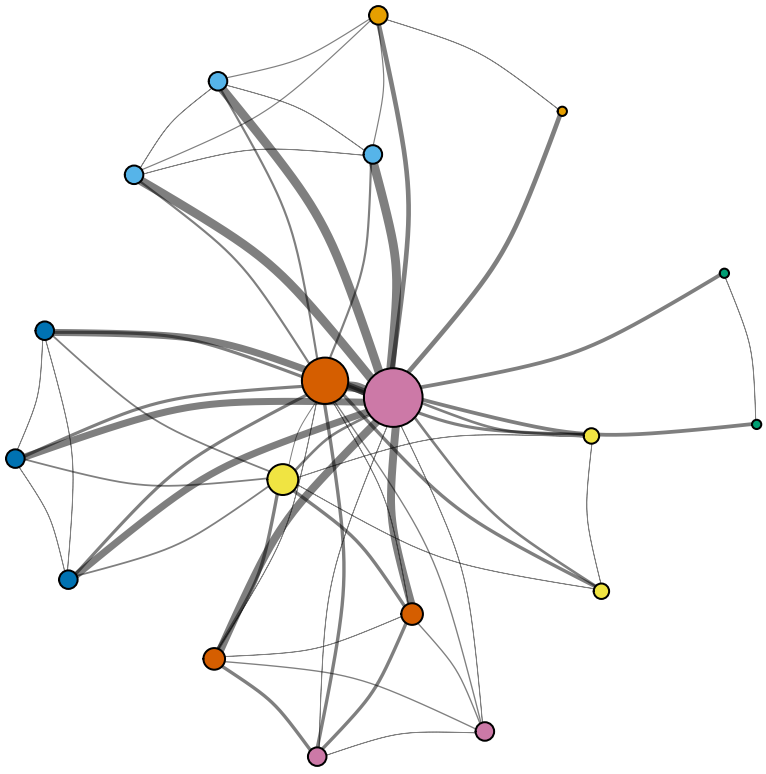


```
g.intra.9.18 <- graph.from.adjacency('C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Da

V(g.intra.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.intra.9.18)
```



9-25

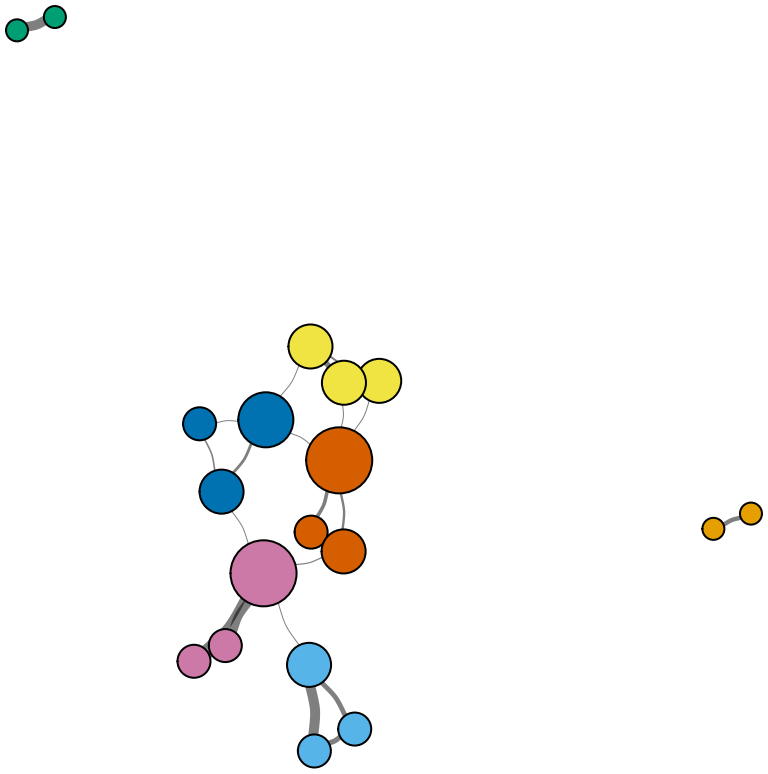
```
g.inter.9.25 = create.graph(file1 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Vi
plot.graph(g.inter.9.25)
```



```
g.intra.9.25 = create.graph(file1 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Vi
                            file2 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Vi
                            nvid1 = 1, nvid2 = 1, method = 2)
```

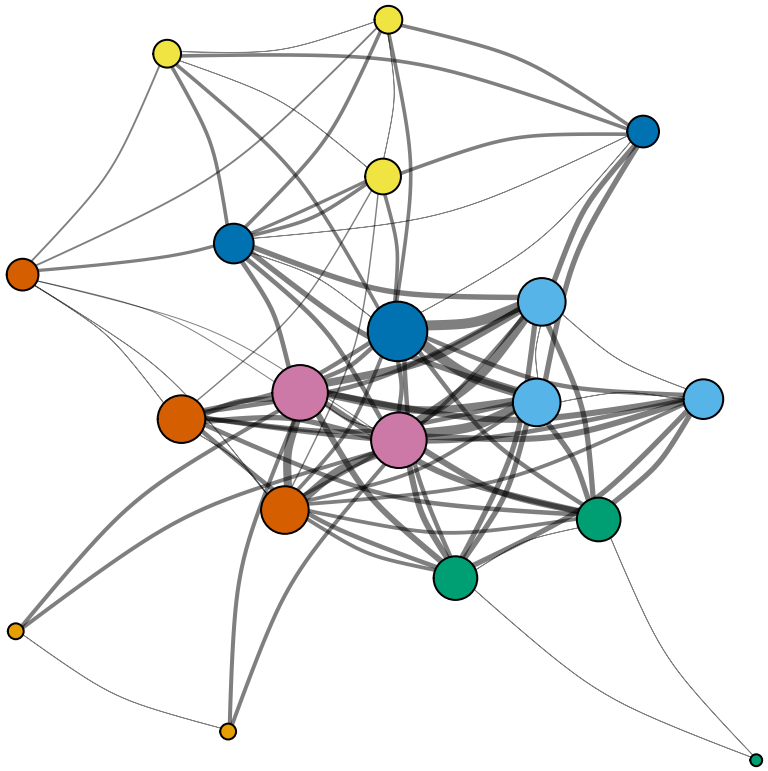
```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
plot.graph(g.intra.9.25)
```



Reliability check

```
g.inter.9.18.DK = create.graph(file1 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/
                               file2 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Da
                               nvid1 = 5, nvid2 = 7, offset1 = 142, method = 1)
plot.graph(g.inter.9.18.DK)
```



```
reliability(g.inter.9.18, g.inter.9.18.DK)
```

```
## * nd.hamming : Hamming distance is originally used for binary networks. The result may not be valid.
```

```
## [1] 0.05847953 0.09805068
```

```
reliability(g.inter.9.18, g.inter.9.18.DK, normalize = TRUE)
```

```
## * nd.hamming : Hamming distance is originally used for binary networks. The result may not be valid.
```

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

```
## [1] 0.8850575 0.7931318
```

```
g.intra.9.18.CW = create.graph(file1 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/
                               file2 = 'C:/Users/Cole/Documents/GRA_Spring2020/Student-Lab-Interactions/Vi
                               nvid1 = 5, nvid2 = 7, method = 2)
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
#g.intra.9.18.sub <- induced.subgraph(g.intra.9.18, c(1, 2, 6, 7, 8, 18, 19))

reliability(g.intra.9.18, igraph::permute(g.intra.9.18.CW,
                                          match(V(g.intra.9.18.CW)$name,
                                                V(g.intra.9.18)$name)), method = 2)
```

```
## * nd.hamming : Hamming distance is originally used for binary networks. The result may not be valid.
```

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

```
reliability(g.intra.9.18, igraph::permute(g.intra.9.18.CW,
                                          match(V(g.intra.9.18.CW)$name,
                                                V(g.intra.9.18)$name)), method = 2, normalize = TRUE)
```

```
## * nd.hamming : Hamming distance is originally used for binary networks. The result may not be valid.
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient
```

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

get legend function

```
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)
}
```

density analysis

```
inter.graphs.wednesday <- list(g.inter.9.11, g.inter.9.18, g.inter.9.25)
intra.graphs.wednesday <- list(g.intra.9.11, g.intra.9.18, g.intra.9.25)

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

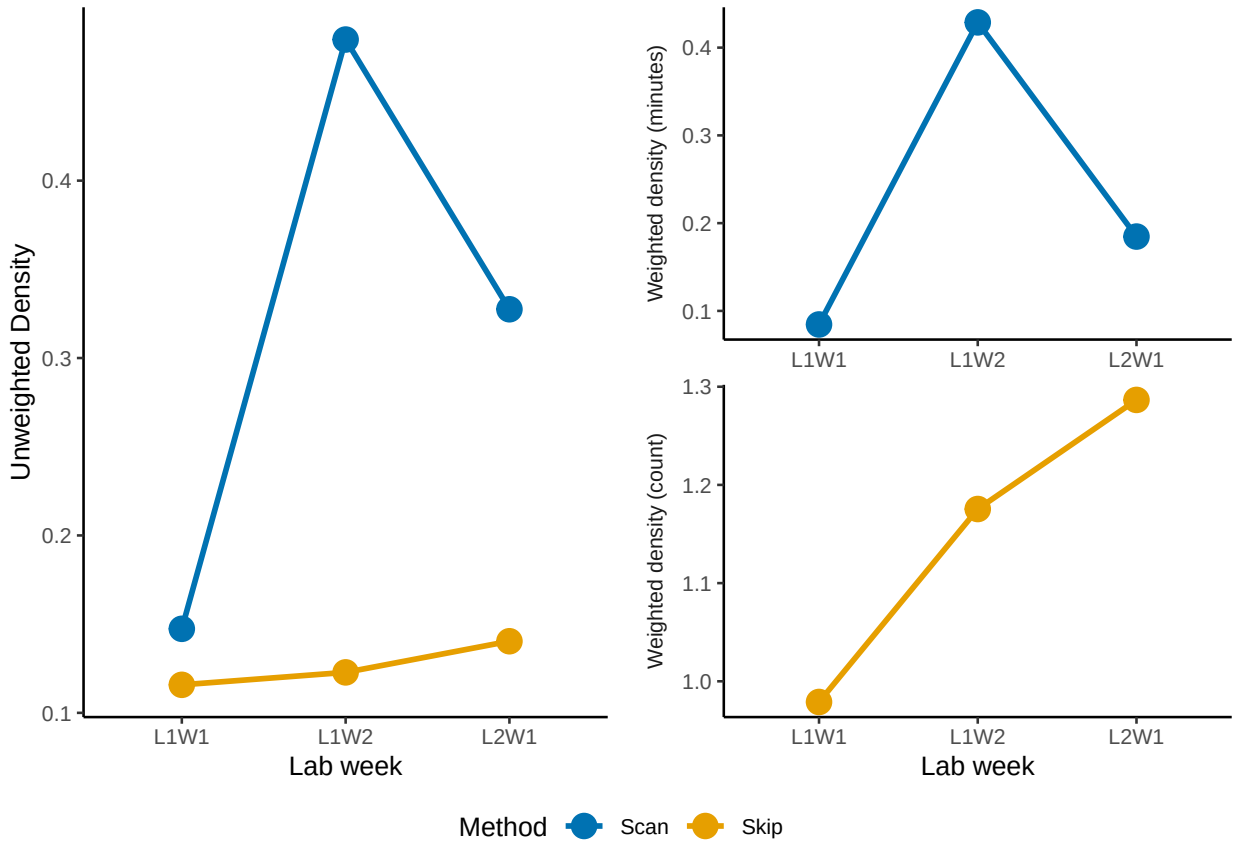
```
## Warning in if (!is.na(var)) {: the condition has length > 1 and only the
## first element will be used
```

```
## Warning in if (!is.na(var)) {: the condition has length > 1 and only the
## first element will be used
```

```
## Warning in if (!is.na(var)) {: the condition has length > 1 and only the
## first element will be used
```

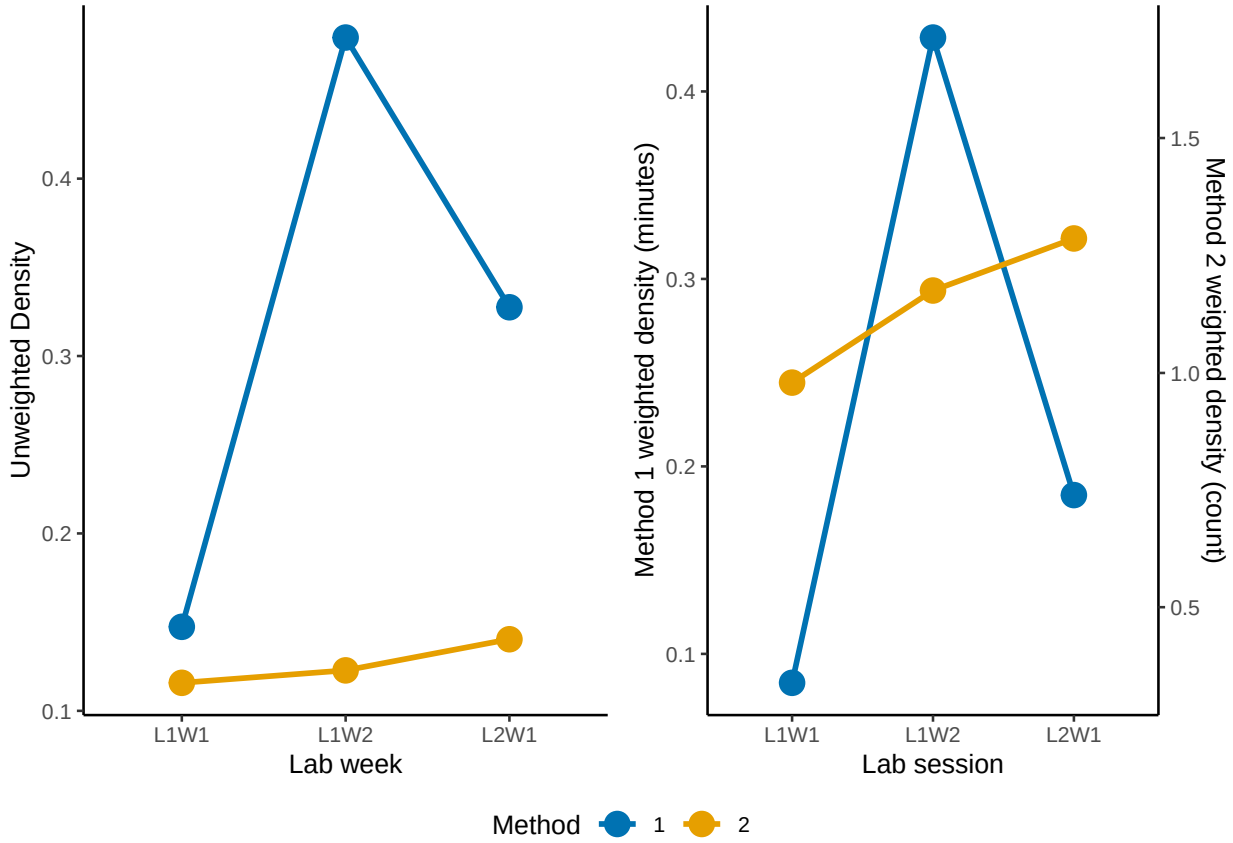
```
## Warning in if (!is.na(var)) {: the condition has length > 1 and only the
## first element will be used
```

```
plot.unweight <- ggplot(df.density %>%
         filter(variable == 'density'),
         aes(x = as.factor(lab.week), y = value, group = method, 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')) +
  # scale_shape_manual(name = 'Lab Number', values = c(16, 15)) +
  theme(legend.position = "none")

plot.weight <- ggplot(df.density %>%
                        filter(variable == 'density.weighted') %>%
                        mutate(method = ifelse(method == 1, 'one', 'two')),
                      aes(x = as.factor(lab.week), y = value, group = method,
                          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 (c
  #facet_wrap(~variable, scales = 'free', labeller = labeller(variable = c('density' = 'Unweighted', 'd
  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), labels = c('A', 'B')) +
  theme(
  strip.background = element_blank(),
  #strip.text.x = element_blank()
  strip.placement = "outside"
) +
  ylab(NULL) +
  theme(legend.position = "bottom")

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))
```



Alternative density graph

```
scale <- max(df.density[(df.density$variable == 'density.weighted') & (df.density$method == 2),
                        'value'])/max(df.density[(df.density$variable == 'density.weighted') &
                                                 (df.density$method == 1), 'value']) + 1

plot.weight <- ggplot(df.density %>%
                        filter(variable == 'density.weighted') %>%
                        mutate(value = ifelse(method == 1, value, value/scale)), aes(x = lab.week, y = 
  geom_point(size = 4) +
  geom_line(stat = 'identity', size = 1) +
  labs(x = 'Lab session', y = 'Method 1 weighted density (minutes)') +
  scale_color_manual(name = 'Method', values = c('#0072B2', '#E69F00')) +
  scale_y_continuous(sec.axis = sec_axis(~.*scale, name = "Method 2 weighted density (count)")) +
  theme(legend.position = "bottom")

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))
```



variance analysis

```
df.vars <- df.create(intra.graphs.wednesday, session = 'Wednesday', lab = c(1, 1), week = c(1, 2),
                     method = 2, func = 'variance') %>%
  mutate(lab.week = paste('L', lab, 'W', week, sep = ''))

ggplot(df.vars, aes(x = lab.week, y = within.variance, group = method)) +
  geom_point(size = 4) +
  geom_line(stat = 'identity', size = 1) +
  labs(x = 'Lab session', y = 'Mean within-group variance')
```