ex-3- ona

Jessica Quansah

2024-04-02

Load libraries needed

Load cleaned up datasets from the starter code provided

```
data_path = "~/GitHub/desktop-tutorial/Exercise-3/"
edges <- read_csv(paste0(data_path,"edges.csv"))
```

```
## Rows: 32906 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): application_number
## dbl  (2): ego_examiner_id, alter_examiner_id
## date (1): advice_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
application <- read_csv(paste0(data_path,"cleaned_applications.csv"))
```

```
## Rows: 2018477 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (11): application_number, examiner_name_last, examiner_name_first, exam...
## dbl   (5): examiner_id, examiner_art_unit, appl_status_code, tc, tenure_days
## date  (5): filing_date, patent_issue_date, abandon_date, earliest_date, late...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Picking Workgroups and Visualizing Demographics

From the data description provided, only 4 of the 9 Technology Centers comprising the agency are included: 1600, 1700, 2100 and 2400. As such I would pick one owrkgroup from each center to allow for some variety. 1600 - Biotechnology and 1700- Materials and Chemical Engineering.

First we will create a column for the workgroup by extracting the first 3 digits. Then we will get the counts per work group in order to pick the one that will give us a good number of data points

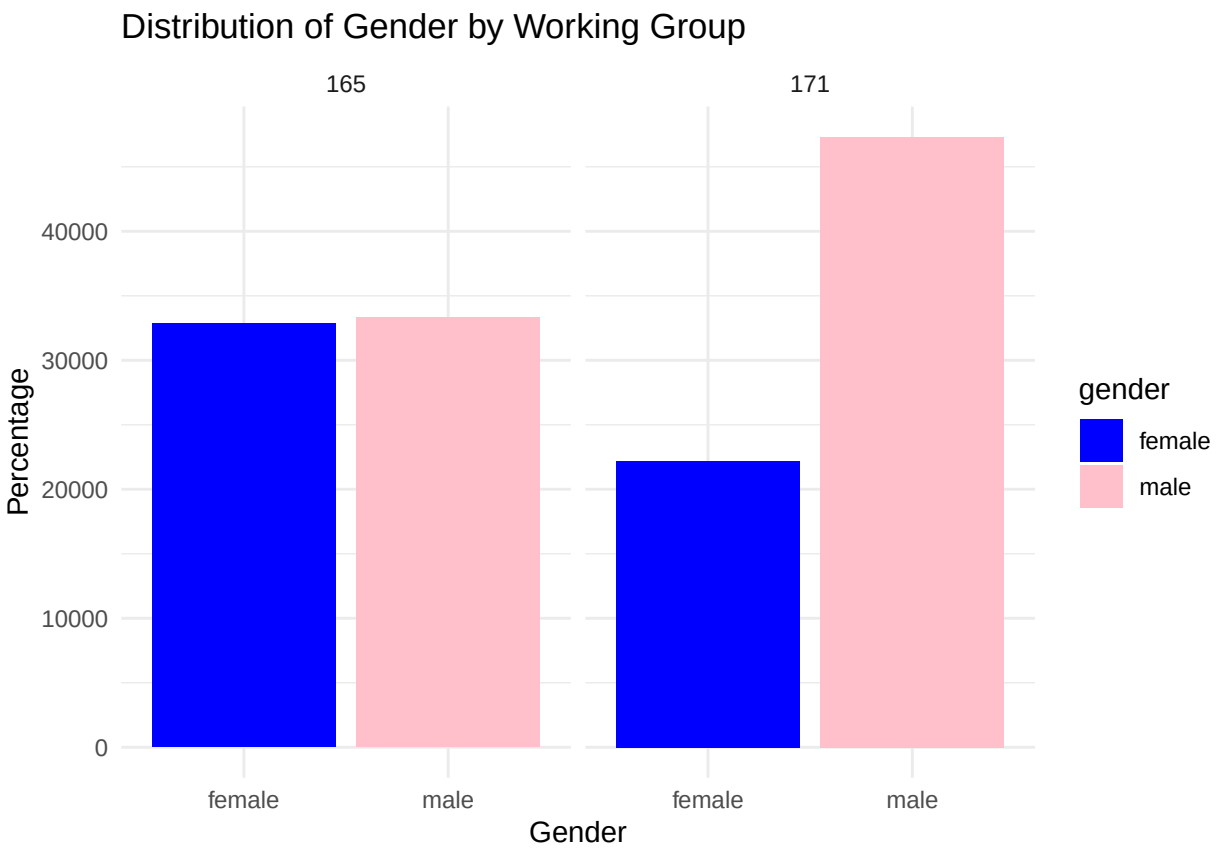
```r
# Extract first three digits of the 4-digit column
application$workgroup <- substr(application$examiner_art_unit, 1, 3)

# Count occurrences of each unique 3-digit code
wgcounts <- table(application$workgroup)

# Print the counts
print(wgcounts)
```

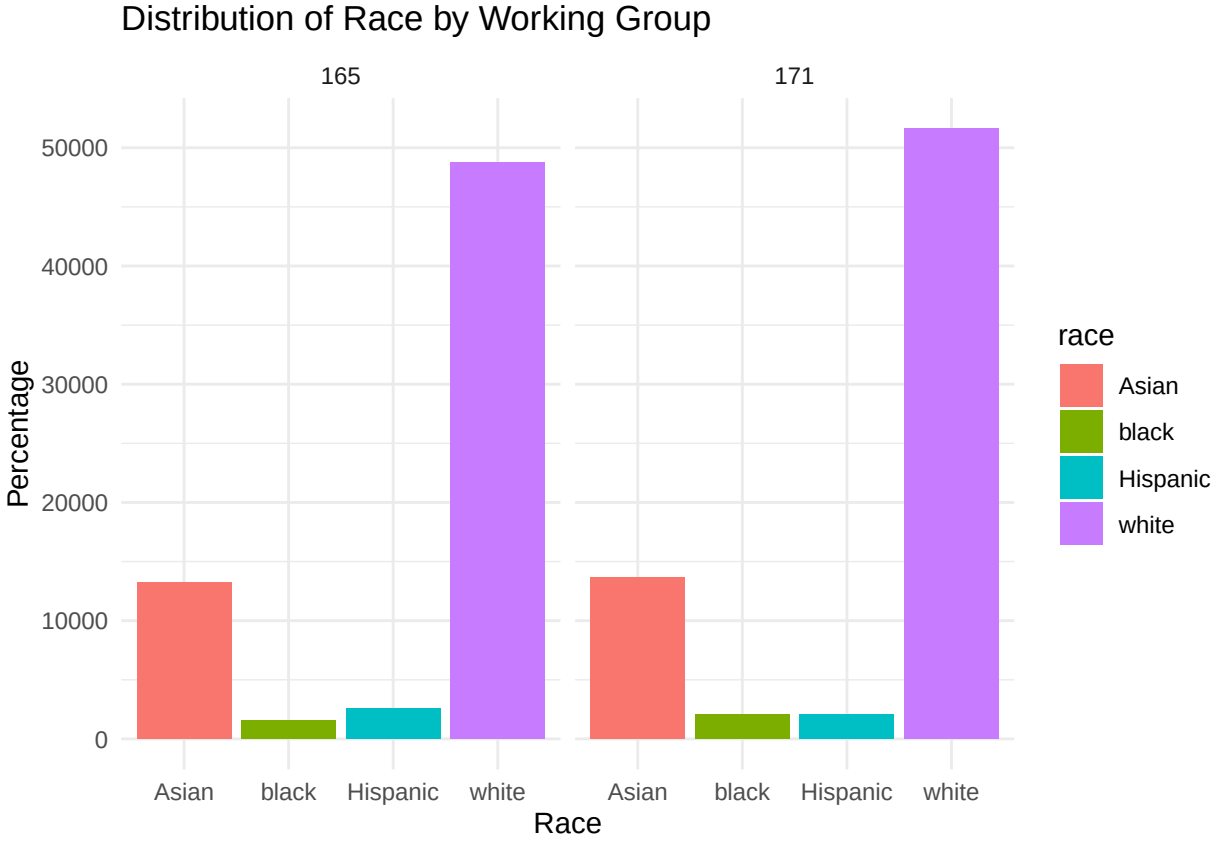
```
##
##   160   161   162   163   164   165   166   167   170   171   172
##   155 89795 141390 90860 93342 75390  8766 35354    45 76544 79195
##   173   174   175   176   177   178   179   210   211   212   213
## 64804 75598 58207 91376 83266 58140 133424    57 60518 52680 30257
##   214   215   216   217   218   219   240   241   242   243   244
## 17964 40229 55780 48772 56974 48047    72 19591 30243 50630 42213
##   245   246   247   248   249
## 42247 54886 48228 41019 22419
```

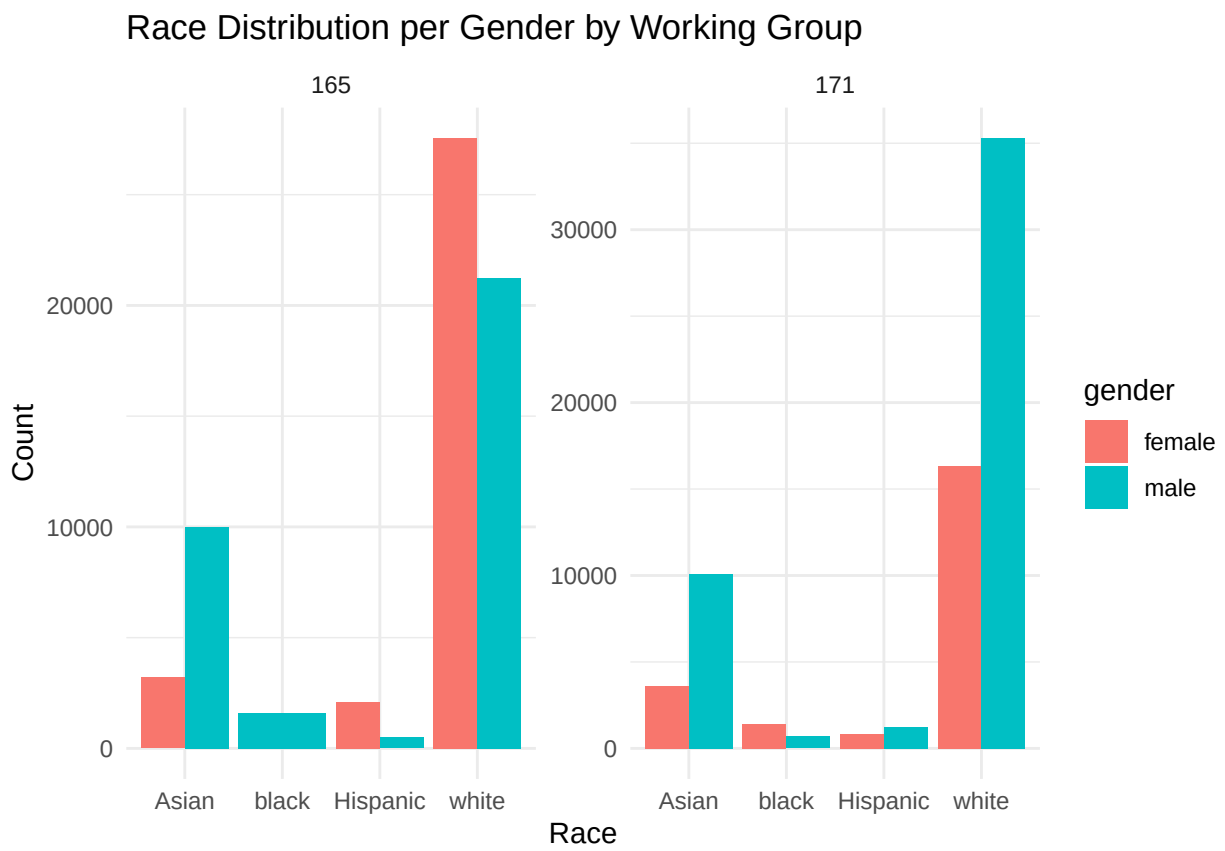
Just based on observation I am going to pick 165 and 171. I wanted both groups to have similar amount of people. I will now filter the rest of the groups

```r
app_subset <- subset(application, workgroup %in% c("165", "171"))
app_subset <- app_subset[!is.na(app_subset$gender), ]
```

Summary Statistics







We can see that in the working group 165, there are more white females whereas in workgroup 171, the dominant race is white males. There are no balck females in working group 165. Separately in both groups the most popular gender is male and the most popular race is white.

Lets look at other summary statistics

```
# Calculate summary statistics by working group
summary_stats <- app_subset %>%
  group_by(workgroup) %>%
  summarise(mean_tenure = mean(tenure_days, na.rm = TRUE),
            median_tenure = median(tenure_days, na.rm = TRUE),
            sd_tenure = sd(tenure_days, na.rm = TRUE),
            .groups = "drop")


# Calculate summary statistics by gender and race within each working group
summary_stats_gender_race <- app_subset %>%
  group_by(workgroup, gender, race) %>%
  summarise(count = n(),  # Count of observations
            mean_tenure = mean(tenure_days, na.rm = TRUE),  # Mean age
            median_tenure = median(tenure_days, na.rm = TRUE),  # Median age
            sd_tenure = sd(tenure_days, na.rm = TRUE),  # Standard deviation of age
            .groups = "drop")

# View the summary statistics
print(summary_stats)
```

```
## # A tibble: 2 x 4
##   workgroup mean_tenure median_tenure sd_tenure
##   <chr>           <dbl>         <dbl>     <dbl>
## 1 165             5814.          6308      904.
## 2 171             5540.          6296     1106.
```

```
print(summary_stats_gender_race)
```

```
## # A tibble: 15 x 7
##    workgroup gender race     count mean_tenure median_tenure sd_tenure
##    <chr>     <chr>  <chr>    <int>       <dbl>         <dbl>     <dbl>
##  1 165       female Asian     3204       4845.          5149     1115.
##  2 165       female Hispanic  2081       6064.          5913      308.
##  3 165       female white    27553       5907.          6308      764.
##  4 165       male   Asian    10014       5970.          6182      572.
##  5 165       male   Hispanic   494       4992.          4200     1009.
##  6 165       male   black     1587       6221.          6329      194.
##  7 165       male   white    21237       5731.          6328     1097.
##  8 171       female Asian     3605       5490.          6330     1146.
##  9 171       female Hispanic   833       6317.          6338      313.
## 10 171       female black     1418       5643.          6345     1104.
## 11 171       female white    16319       5562.          6312     1225.
## 12 171       male   Asian    10080       5431.          5625     1098.
## 13 171       male   Hispanic  1248       4532.          4403     1316.
## 14 171       male   black      685       6339           6339        0
## 15 171       male   white    35296       5562.          6283     1026.
```

IN work group 165, median and mean trnure is significantly higher for black males but for workgroup 171, Hispanic females have high tenures.

Creating Advice Networks

The goal here was to create nodes and network data then visualize for each platform.Then identify the examiner's that have high betweenness centrality, high degree centrality and what characteristics these people exhibit. But as the time of submission of this assignment, I had a challenge with mainly getting error messages indicating duplicates or missing values in either my nodes or edges dataset but after removing my duplicates and missing values, I then get an error message around negative or invalid values - "Error in add_vertices(gr, nrow(nodes) - gorder(gr)) : At vendor/cigraph/src/graph/type_indexededgelist.c:388 : Cannot add negative number of vertices. Invalid value" Which I was not able to trouble shoot prior to the submission of this file as such this was my final diagrams

```
# Step 1: Preprocessing applications data
# Filter applications data for relevant workgroup
workgroup1 <- "165"
workgroup2 <- "171"
workgroup1_data <- app_subset %>% filter(workgroup == workgroup1)
workgroup2_data <- app_subset %>% filter(workgroup == workgroup2)

# Create nodes for department1
nodes_workgroup1 <- workgroup1_data %>%
  select(examiner_id, gender, race, tenure_days) %>%
  rename(node = examiner_id)
```

```
# Create nodes for department2
nodes_workgroup2 <- workgroup2_data %>%
  select(examiner_id, gender, race, tenure_days) %>%
  rename(node = examiner_id)

# Step 2: Preprocessing edges data
# Assuming your edges data frame is named 'edges'

# Filter edges data for relevant departments
edges_workgroup1 <- edges %>%
  filter(ego_examiner_id %in% workgroup1_data$examiner_id)
edges_workgroup2 <- edges %>%
  filter(ego_examiner_id %in% workgroup2_data$examiner_id)


# Create graph for department1
graph_165 <- graph_from_data_frame(edges_workgroup1, vertices = nodes_workgroup1, directed = FALSE)

# Create graph for department2
graph_171 <- graph_from_data_frame(edges_workgroup2, vertices = nodes_workgroup2, directed = FALSE)
```

##Looking at some centrality Measures

```
# Compute betweenness centrality for department1
betweenness_165 <- betweenness(graph_165, directed = FALSE)

# Compute closeness centrality for department1
closeness_165 <- closeness(graph_165, normalized = TRUE)

# Compute degree centrality for department1
degree_165 <- degree(graph_165)

# Combine centrality measures with node attributes
nodes_workgroup1$betweenness <- betweenness_165
nodes_workgroup1$closeness <- closeness_165
nodes_workgroup1$degree <- degree_165

# Repeat the above steps for department2
betweenness_171 <- betweenness(graph_171, directed = FALSE)
closeness_171 <- closeness(graph_171, normalized = TRUE)
degree_171 <- degree(graph_171)
nodes_workgroup2$betweenness <- betweenness_171
nodes_workgroup2$closeness <- closeness_171
nodes_workgroup2$degree <- degree_171

# Print top 5 nodes for department1
cat("Top 5 nodes for department1 based on betweenness centrality:\n")
top_betweenness_165 <- head(nodes_workgroup1[order(-nodes_workgroup1$betweenness), ], 5)
print(top_betweenness_165[, c("examiner_id", "gender", "race", "betweenness")])

cat("\nTop 5 nodes for department1 based on closeness centrality:\n")
top_closeness_165 <- head(nodes_workgroup1[order(-nodes_workgroup1$closeness), ], 5)
print(top_closeness_165[, c("examiner_id", "gender", "race", "closeness")])
```

```
cat("\nTop 5 nodes for department1 based on degree centrality:\n")
top_degree_165 <- head(nodes_workgroup1[order(-nodes_workgroup1$degree), ], 5)
print(top_degree_165[, c("examiner_id", "gender", "race", "degree")])

# Repeat for department2
cat("\nTop 5 nodes for department2 based on betweenness centrality:\n")
top_betweenness_171 <- head(nodes_workgroup2[order(-nodes_workgroup2$betweenness), ], 5)
print(top_betweenness_171[, c("examiner_id", "gender", "race", "betweenness")])

cat("\nTop 5 nodes for department2 based on closeness centrality:\n")
top_closeness_171 <- head(nodes_workgroup2[order(-nodes_workgroup2$closeness), ], 5)
print(top_closeness_171[, c("examiner_id", "gender", "race", "closeness")])

cat("\nTop 5 nodes for department2 based on degree centrality:\n")
top_degree_171 <- head(nodes_workgroup2[order(-nodes_workgroup2$degree), ], 5)
print(top_degree_171[, c("examiner_id", "gender", "race", "degree")])
```