Exercise 3 Amr Maraqa

Load data

Load the following data: + applications from `app_data_sample.parquet` + edges from `edges_sample.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.
```

```
## # A tibble: 2,018,477 x 16
##    applicat~1 filing_d~2 exami~3 exami~4 exami~5 exami~6 exami~7 uspc_~8 uspc_~9
##    <chr>      <date>     <chr>   <chr>   <chr>     <dbl>   <dbl> <chr>   <chr>
##  1 08284457   2000-01-26 HOWARD  JACQUE~ V         96082    1764 508     273000
##  2 08413193   2000-10-11 YILDIR~ BEKIR   L         87678    1764 208     179000
##  3 08531853   2000-05-17 HAMILT~ CYNTHIA <NA>      63213    1752 430     271100
##  4 08637752   2001-07-20 MOSHER  MARY    <NA>      73788    1648 530     388300
##  5 08682726   2000-04-10 BARR    MICHAEL E         77294    1762 427     430100
##  6 08687412   2000-04-28 GRAY    LINDA   LAMEY     68606    1734 156     204000
##  7 08716371   2004-01-26 MCMILL~ KARA    RENITA    89557    1627 424     401000
##  8 08765941   2000-06-23 FORD    VANESSA L         97543    1645 424     001210
##  9 08776818   2000-02-04 STRZEL~ TERESA  E         98714    1637 435     006000
## 10 08809677   2002-02-20 KIM     SUN     U         65530    1723 210     645000
## # ... with 2,018,467 more rows, 7 more variables: patent_number <chr>,
## #   patent_issue_date <date>, abandon_date <date>, disposal_type <chr>,
## #   appl_status_code <dbl>, appl_status_date <chr>, tc <dbl>, and abbreviated
## #   variable names 1: application_number, 2: filing_date,
## #   3: examiner_name_last, 4: examiner_name_first, 5: examiner_name_middle,
## #   6: examiner_id, 7: examiner_art_unit, 8: uspc_class, 9: uspc_subclass
```

```
## # A tibble: 32,906 x 4
##    application_number advice_date ego_examiner_id alter_examiner_id
##    <chr>              <date>                <dbl>             <dbl>
##  1 09402488           2008-11-17            84356             66266
##  2 09402488           2008-11-17            84356             63519
##  3 09402488           2008-11-17            84356             98531
##  4 09445135           2008-08-21            92953             71313
##  5 09445135           2008-08-21            92953             93865
##  6 09445135           2008-08-21            92953             91818
##  7 09479304           2008-12-15            61767             69277
##  8 09479304           2008-12-15            61767             92446
```

```
##  9 09479304           2008-12-15            61767             66805
## 10 09479304           2008-12-15            61767             70919
## # ... with 32,896 more rows
```

Get gender for examiners

We'll get gender based on the first name of the examiner, which is recorded in the field `examiner_name_first`. We'll use library `gender` for that, relying on a modified version of their own example.

Note that there are over 2 million records in the applications table – that's because there are many records for each examiner, as many as the number of applications that examiner worked on during this time frame. Our first step therefore is to get all *unique* names in a separate list `examiner_names`. We will then guess gender for each one and will join this table back to the original dataset. So, let's get names without repetition:

```
## # A tibble: 2,595 x 1
##    examiner_name_first
##    <chr>
##  1 JACQUELINE
##  2 BEKIR
##  3 CYNTHIA
##  4 MARY
##  5 MICHAEL
##  6 LINDA
##  7 KARA
##  8 VANESSA
##  9 TERESA
## 10 SUN
## # ... with 2,585 more rows
```

Now let's use function `gender()` as shown in the example for the package to attach a gender and probability to each name and put the results into the table `examiner_names_gender`

```
## # A tibble: 1,822 x 3
##    examiner_name_first gender proportion_female
##    <chr>               <chr>              <dbl>
##  1 AARON               male              0.0082
##  2 ABDEL               male              0
##  3 ABDOU               male              0
##  4 ABDUL               male              0
##  5 ABDULHAKIM          male              0
##  6 ABDULLAH            male              0
##  7 ABDULLAHI           male              0
##  8 ABIGAIL             female            0.998
##  9 ABIMBOLA            female            0.944
## 10 ABRAHAM             male              0.0031
## # ... with 1,812 more rows
```

Finally, let's join that table back to our original applications data and discard the temporary tables we have just created to reduce clutter in our environment.

```
##            used (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  5054934  270   14250452  761.1  14250452  761.1
## Vcells 56873881  434  134532627 1026.5 134516664 1026.3
```

Guess the examiner's race

We'll now use package `wru` to estimate likely race of an examiner. Just like with gender, we'll get a list of unique names first, only now we are using surnames.

```
## # A tibble: 3,806 x 1
##    surname
##    <chr>
##  1 HOWARD
##  2 YILDIRIM
##  3 HAMILTON
##  4 MOSHER
##  5 BARR
##  6 GRAY
##  7 MCMILLIAN
##  8 FORD
##  9 STRZELECKA
## 10 KIM
## # ... with 3,796 more rows
```

We'll follow the instructions for the package outlined here https://github.com/kosukeimai/wru.

```
## Warning: Unknown or uninitialised column: 'state'.
```

```
## Proceeding with last name predictions...
```

```
## i All local files already up-to-date!
## i All local files already up-to-date!
```

```
## 701 (18.4%) individuals' last names were not matched.
```

```
## # A tibble: 3,806 x 6
##    surname    pred.whi pred.bla pred.his pred.asi pred.oth
##    <chr>         <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
##  1 HOWARD       0.597   0.295    0.0275   0.00690   0.0741
##  2 YILDIRIM     0.807   0.0273   0.0694   0.0165    0.0798
##  3 HAMILTON     0.656   0.239    0.0286   0.00750   0.0692
##  4 MOSHER       0.915   0.00425  0.0291   0.00917   0.0427
##  5 BARR         0.784   0.120    0.0268   0.00830   0.0615
##  6 GRAY         0.640   0.252    0.0281   0.00748   0.0724
##  7 MCMILLIAN    0.322   0.554    0.0212   0.00340   0.0995
##  8 FORD         0.576   0.320    0.0275   0.00621   0.0697
##  9 STRZELECKA   0.472   0.171    0.220    0.0825    0.0543
## 10 KIM          0.0169  0.00282  0.00546  0.943     0.0319
## # ... with 3,796 more rows
```

As you can see, we get probabilities across five broad US Census categories: white, black, Hispanic, Asian and other. (Some of you may correctly point out that Hispanic is not a race category in the US Census, but these are the limitations of this package.)

Our final step here is to pick the race category that has the highest probability for each last name and then join the table back to the main applications table. See this example for comparing values across columns: https://www.tidyverse.org/blog/2020/04/dplyr-1-0-0-rowwise/. And this one for `case_when()` function: https://dplyr.tidyverse.org/reference/case_when.html.

```
## # A tibble: 3,806 x 8
##    surname    pred.whi pred.bla pred.his pred.asi pred.oth max_race_p race
##    <chr>         <dbl>    <dbl>    <dbl>    <dbl>    <dbl>      <dbl> <chr>
##  1 HOWARD       0.597   0.295    0.0275   0.00690   0.0741      0.597 white
##  2 YILDIRIM     0.807   0.0273   0.0694   0.0165    0.0798      0.807 white
##  3 HAMILTON     0.656   0.239    0.0286   0.00750   0.0692      0.656 white
##  4 MOSHER       0.915   0.00425  0.0291   0.00917   0.0427      0.915 white
##  5 BARR         0.784   0.120    0.0268   0.00830   0.0615      0.784 white
##  6 GRAY         0.640   0.252    0.0281   0.00748   0.0724      0.640 white
##  7 MCMILLIAN    0.322   0.554    0.0212   0.00340   0.0995      0.554 black
##  8 FORD         0.576   0.320    0.0275   0.00621   0.0697      0.576 white
##  9 STRZELECKA   0.472   0.171    0.220    0.0825    0.0543      0.472 white
## 10 KIM          0.0169  0.00282  0.00546  0.943     0.0319      0.943 Asian
## # ... with 3,796 more rows
```

Let's join the data back to the applications table.

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  5054893 270.0   14250452  761.1  14250452  761.1
## Vcells 58892652 449.4  134532627 1026.5 134516664 1026.3
```

Examiner's tenure

To figure out the timespan for which we observe each examiner in the applications data, let's find the first and the last observed date for each examiner. We'll first get examiner IDs and application dates in a separate table, for ease of manipulation. We'll keep examiner ID (the field `examiner_id`), and earliest and latest dates for each application (`filing_date` and `appl_status_date` respectively). We'll use functions in package `lubridate` to work with date and time values.

```
## # A tibble: 2,018,477 x 3
##    examiner_id filing_date appl_status_date
##          <dbl> <date>      <chr>
##  1       96082 2000-01-26  30jan2003 00:00:00
##  2       87678 2000-10-11  27sep2010 00:00:00
##  3       63213 2000-05-17  30mar2009 00:00:00
##  4       73788 2001-07-20  07sep2009 00:00:00
##  5       77294 2000-04-10  19apr2001 00:00:00
##  6       68606 2000-04-28  16jul2001 00:00:00
##  7       89557 2004-01-26  15may2017 00:00:00
##  8       97543 2000-06-23  03apr2002 00:00:00
##  9       98714 2000-02-04  27nov2002 00:00:00
## 10       65530 2002-02-20  23mar2009 00:00:00
## # ... with 2,018,467 more rows
```

The dates look inconsistent in terms of formatting. Let's make them consistent. We'll create new variables `start_date` and `end_date`.

Let's now identify the earliest and the latest date for each examiner and calculate the difference in days, which is their tenure in the organization.

```
## # A tibble: 5,625 x 6
##    examiner_id earliest_date latest_date tenure_days tenure_years tenure
```

```
##          <dbl> <date>        <date>             <dbl>        <dbl> <chr>
##  1       59012 2004-07-28    2015-07-24          4013        11.0  10-14
##  2       59025 2009-10-26    2017-05-18          2761         7.56 6-9
##  3       59030 2005-12-12    2017-05-22          4179        11.4  10-14
##  4       59040 2007-09-11    2017-05-23          3542         9.70 10-14
##  5       59052 2001-08-21    2007-02-28          2017         5.53 6-9
##  6       59054 2000-11-10    2016-12-23          5887        16.1  15+
##  7       59055 2004-11-02    2007-12-26          1149         3.15 3-5
##  8       59056 2000-03-24    2017-05-22          6268        17.2  15+
##  9       59074 2000-01-31    2017-03-17          6255        17.1  15+
## 10       59081 2011-04-21    2017-05-19          2220         6.08 6-9
## # ... with 5,615 more rows
```

Joining back to the applications data.

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  5054850 270.0   14250452  761.1  14250452  761.1
## Vcells 68985012 526.4  134532627 1026.5 134516664 1026.3
```

Creating work group column and dropping NAs

Finding the work group numbers of each art unit and creating a data set for work groups 213 and 174

Getting summary statistics for work groups

```
## Joining, by = "examiner_id"
```

```
## Joining, by = "examiner_gender"
## Joining, by = "examiner_gender"
## Joining, by = "examiner_race"
## Joining, by = "examiner_race"
```

Summary Statistics

Work groups 213 and 174 are evaluated.

- Work group 213 has 262 employees, and 174 has 252
- Work group 213 is more male dominated with 78.2% vs 65.5% for 174.
- Both work groups consist predominantly of white examiners, with Asian examiners constituting the second biggest race group.
- The average tenure among examiners in Work group 213 is lower than that of the examiners in 174. This means that 213 has bigger groups of less experienced examiners, which may also mean that examiners in 213 are younger, on average.

Table 1: Total Number of Examiners

examiner_workgroup	n
174	252
213	262

Table 2: Gender Distribution

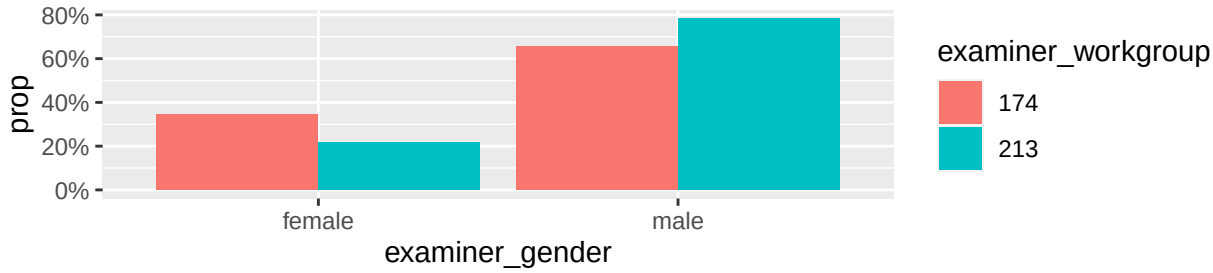
examiner_workgroup	female	male
174	34.52	65.48
213	21.76	78.24

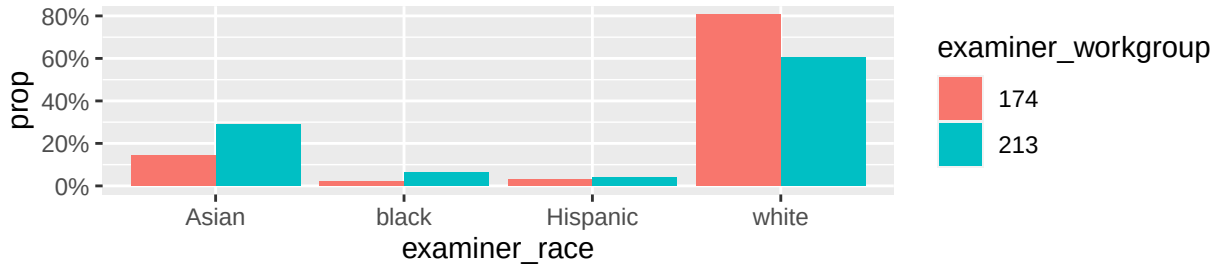
Table 3: Race Distribution

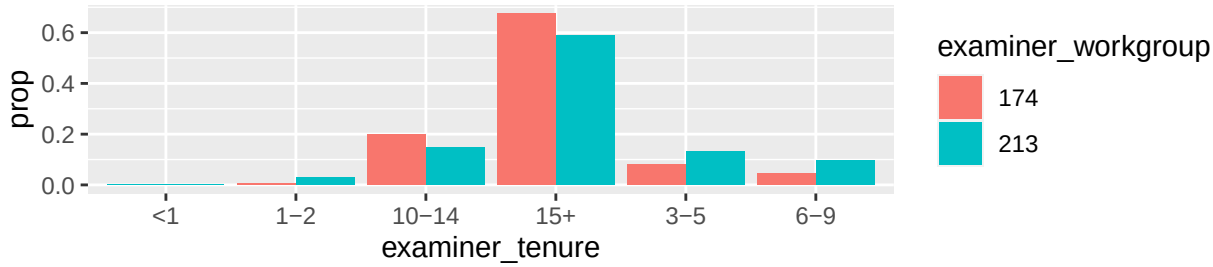
examiner_workgroup	Asian	black	Hispanic	white
174	14.29	1.98	3.17	80.56
213	29.01	6.49	3.82	60.69

Table 4: Tenure Distribution

examiner_workgroup	1-2	10-14	15+	3-5	6-9	<1
174	0.40	19.84	67.46	7.94	4.37	NA
213	3.05	14.89	58.78	13.36	9.54	0.38





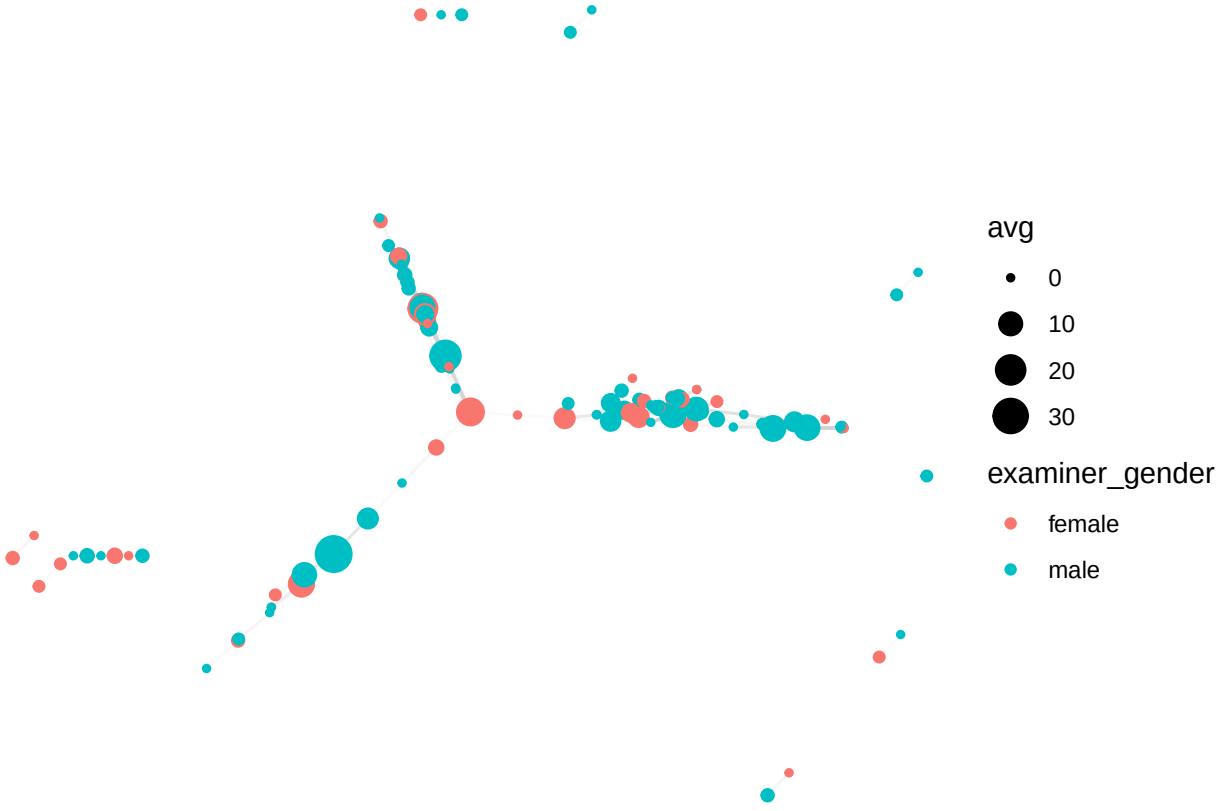


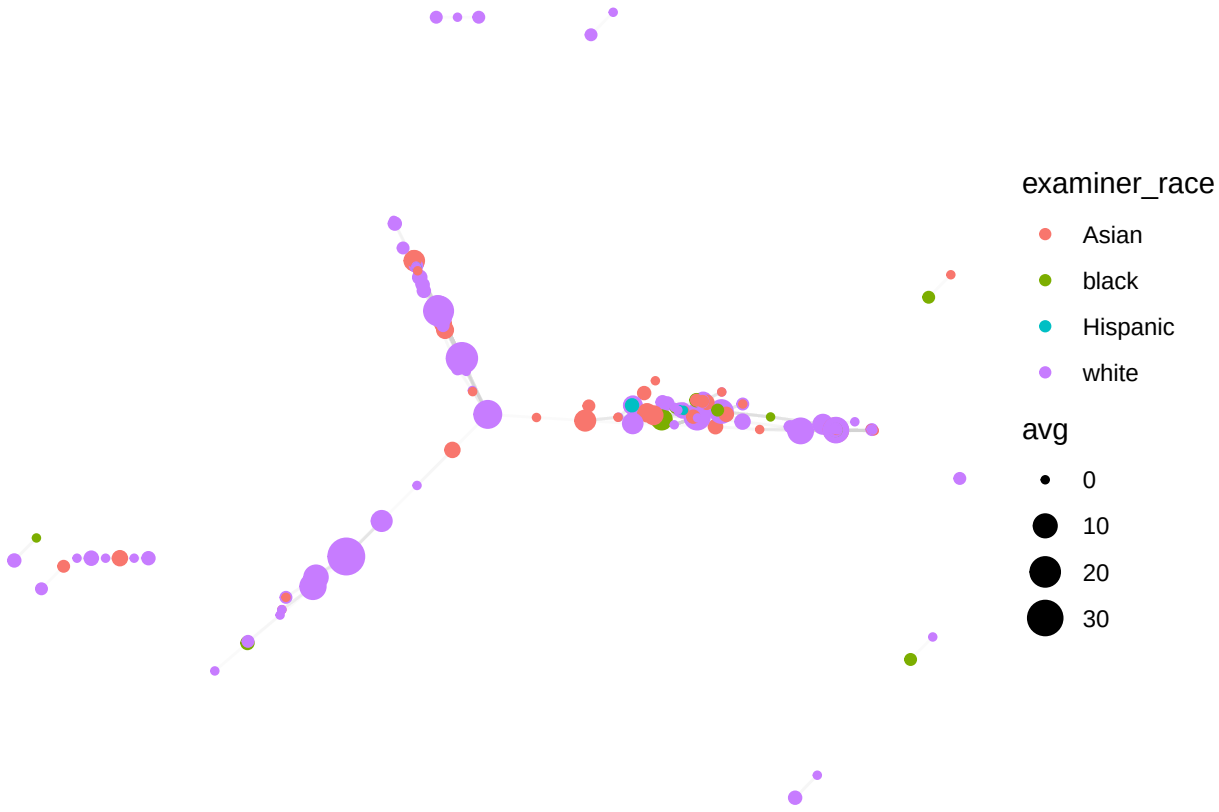
Network Visualization

There seems to be three groups which are connected through one node. This is critical, as the absence of this examiner, would render communication between these clusters impossible. Given that these clusters

originate from two work groups, it can deduced that there is a clear separation between examiners in one of them, which can render it dysfunctional should the connecting examiner leave. This is a sign to the the USPTO that teamwork within and between these two work groups is at jeopardy due to the centralization of the communication.

Within each of the 3 clusters, no segregation by gender or by race appears to exist. This could be due to the non-dominant groups being too small to form their own cluster, to the fact that the employees are interested in maintaining diverse groups.





Discussion

Gender: The difference in gender doesn't seem to impact the popularity of an examiner in and across the 213 and 174 work groups; the measures of degree centrality and betweenness are very similar for both genders.

Race: On the other hand, there exists a clear discrepancy between the various races within the work groups. White people hold the most pivotal positions within the network and are most frequently the channels of communication between examiners. Meanwhile, black people seem to always be on the ends of the networks or aren't detrimental to their flow of information. As for popularity, White and Asian examiners are very similar. It's hard to attribute these differences to racial discrimination, however, as the white community is over-represented in these work groups. The problem may be in the systematic racism in the recruitment process of the UPSTO, but no conclusions can be drawn about the examiners and the relationships between them.

Table 5: Gender Centrality Scores

examiner_gender	top10_degree	top10_bet	mean_degree	mean_bet
female	18.00	12.50	3.152174	1.130435
male	16.33	13.55	3.032000	1.544000

examiner_race	top10_degree	top10_bet	mean_degree	mean_bet
Asian	15.50	3.67	3.289474	0.2894737
black	13.00	0.00	2.636364	0.0000000

examiner_race	top10_degree	top10_bet	mean_degree	mean_bet
white	17.55	19.88	3.118644	1.9830508