Exercise4

```
warnings()
```

Libraries

```
library(ggraph)
library(igraph)

library(arrow)
library(tidyverse)
library(gender)
library(wru)
library(lubridate)

library(ggplot2)
library(gridExtra)
library(grid)
```

Data

```
data_path <- "Data/"
applications <- read_parquet(paste0(data_path,"app_data_sample.parquet"))
edges <- read_csv(paste0(data_path,"edges_sample.csv"))
```

Add gender

```
# get a list of first names without repetitions
examiner_names <- applications %>%
  distinct(examiner_name_first)

# get a table of names and gender
examiner_names_gender <- examiner_names %>%
  do(results = gender(.$examiner_name_first, method = "ssa")) %>%
  unnest(cols = c(results), keep_empty = TRUE) %>%
  select(
    examiner_name_first = name,
    gender,
    proportion_female
  )
# remove extra columns from the gender table
```

```
examiner_names_gender <- examiner_names_gender %>%
  select(examiner_name_first, gender)

# joining gender back to the dataset
applications <- applications %>%
  left_join(examiner_names_gender, by = "examiner_name_first")

# cleaning up
rm(examiner_names)
rm(examiner_names_gender)
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  4712508 251.7    8244992 440.4  4731964 252.8
## Vcells 49747373 379.6   95707868 730.2 80062902 610.9
```

Add race

```
# get list of distinct last names
examiner_surnames <- applications %>%
  select(surname = examiner_name_last) %>%
  distinct()

examiner_race <- predict_race(voter.file = examiner_surnames, surname.only = T) %>%
  as_tibble()
```

```
## [1] "Proceeding with surname-only predictions..."
```

```
# infer racial probabilities from surname tibble
examiner_race <- examiner_race %>%
  mutate(max_race_p = pmax(pred.asi, pred.bla, pred.his, pred.oth, pred.whi)) %>%
  mutate(race = case_when(
    max_race_p == pred.asi ~ "Asian",
    max_race_p == pred.bla ~ "black",
    max_race_p == pred.his ~ "Hispanic",
    max_race_p == pred.oth ~ "other",
    max_race_p == pred.whi ~ "white",
    TRUE ~ NA_character_
  ))

# removing extra columns and merge into application data
examiner_race <- examiner_race %>%
  select(surname,race)

applications <- applications %>%
  left_join(examiner_race, by = c("examiner_name_last" = "surname"))
# cleanup
rm(examiner_race)
rm(examiner_surnames)
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  5052519 269.9    8244992 440.4  5692642 304.1
## Vcells 53433930 407.7   95707868 730.2 94221725 718.9
```

Add tenure

```
# get all application filing dates
examiner_dates <- applications %>%
  select(examiner_id, filing_date, appl_status_date)

# calculate start and end date from filing / status date respectively
examiner_dates <- examiner_dates %>%
  mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))

# for each examiner, get earliest and latest days, then interval between them as tenure in days
examiner_dates <- examiner_dates %>%
  group_by(examiner_id) %>%
  summarise(
    earliest_date = min(start_date, na.rm = TRUE),
    latest_date = max(end_date, na.rm = TRUE),
    tenure_days = interval(earliest_date, latest_date) %/% days(1)
    ) %>%
  filter(year(latest_date)<2018)

# merge and clean
applications <- applications %>%
  left_join(examiner_dates, by = "examiner_id")

rm(examiner_dates)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  5066569 270.6   14718374  786.1  14718374  786.1
## Vcells 65812690 502.2  137995329 1052.9 137870790 1051.9
```

Add application duration

```
# get all application filing dates
application_dates <- applications %>%
  select(application_number, filing_date, appl_status_date)

# calculate start and end date from filing / status date respectively
application_dates <- application_dates %>%
  mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))

# for each application, get earliest and latest days, then interval between them as appl days
application_dates <- application_dates %>%
  summarise(
    application_number = application_number,
    filed = start_date,
```

```
    decision = end_date,
    appl_days = interval(filed, decision) %/% days(1)
  ) %>%
  filter(year(decision)<2018)

# merge and clean
applications <- applications %>%
  left_join(application_dates, by = "application_number")

rm(application_dates)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  5066570 270.6   14718374  786.1  14718374  786.1
## Vcells 71868107 548.4  202368509 1544.0 201867132 1540.2
```

Check completeness of the dataset to this point

```
library(skimr)
applications %>% skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	2018477
Number of columns	24
Column type frequency:	
character	11
Date	7
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
application_number	0	1.00	8	8	0	2018477	0
examiner_name_last	0	1.00	2	17	0	3806	0
examiner_name_first	0	1.00	1	12	0	2595	0
examiner_name_middle	471770	0.77	1	12	0	515	0
uspc_class	4	1.00	3	3	0	416	0
uspc_subclass	1677	1.00	6	6	0	6154	0
patent_number	931651	0.54	4	7	0	1086824	0
disposal_type	0	1.00	3	4	0	3	0
appl_status_date	4610	1.00	18	18	0	5705	0
gender	303859	0.85	4	6	0	2	0
race	0	1.00	5	8	0	5	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
filing_date	0	1.00	2000-01-02	2017-05-26	2009-07-23	6204
patent_issue_date	931178	0.54	1997-03-04	2017-06-20	2012-05-22	891
abandon_date	1417057	0.30	1965-07-20	2050-06-30	2011-04-19	5052
earliest_date	25863	0.99	2000-01-02	2016-03-03	2000-08-31	2322
latest_date	25863	0.99	2000-09-14	2017-12-06	2017-05-20	870
filed	4634	1.00	2000-01-02	2017-05-25	2009-07-15	6200
decision	4634	1.00	2000-03-20	2017-12-06	2013-11-20	5687

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
examiner_id	9229	1.00	78712.39	13606.61	59012	66476	75243	93754	99990	
examiner_art_unit	0	1.00	1928.02	304.38	1600	1671	1773	2171	2498	
appl_status_code	4609	1.00	145.94	51.72	1	150	150	161	865	
tc	0	1.00	1876.91	298.82	1600	1600	1700	2100	2400	
tenure_days	25863	0.99	5524.85	1102.45	27	4955	6076	6335	6518	
appl_days	4634	1.00	1354.52	999.63	-1228	735	1116	1660	6332	

Given that our goal is to measure the relationship between centrality and application processing time, there are a few variables here that may be worth imputing to remove NaNs.

- Gender
- tenure days
- appl days

We will use R's mice package which performs multiple imputation under the assumption that any missing data is 'Missing At Random' ie the probability that a value is missing depends only on the observed value itself. Mice will impute data for each input variable by specifying a unique imputation model per-variable. Ie if our feature set consists of X1, X2, … Xn and X1 has missing values, it will be imputed based on the patterns observed in X2….Xn.

Before we do this, we have to remove some variables which may be missing not-at-random, or are deemed to be unhelpful for the later modelling stage.

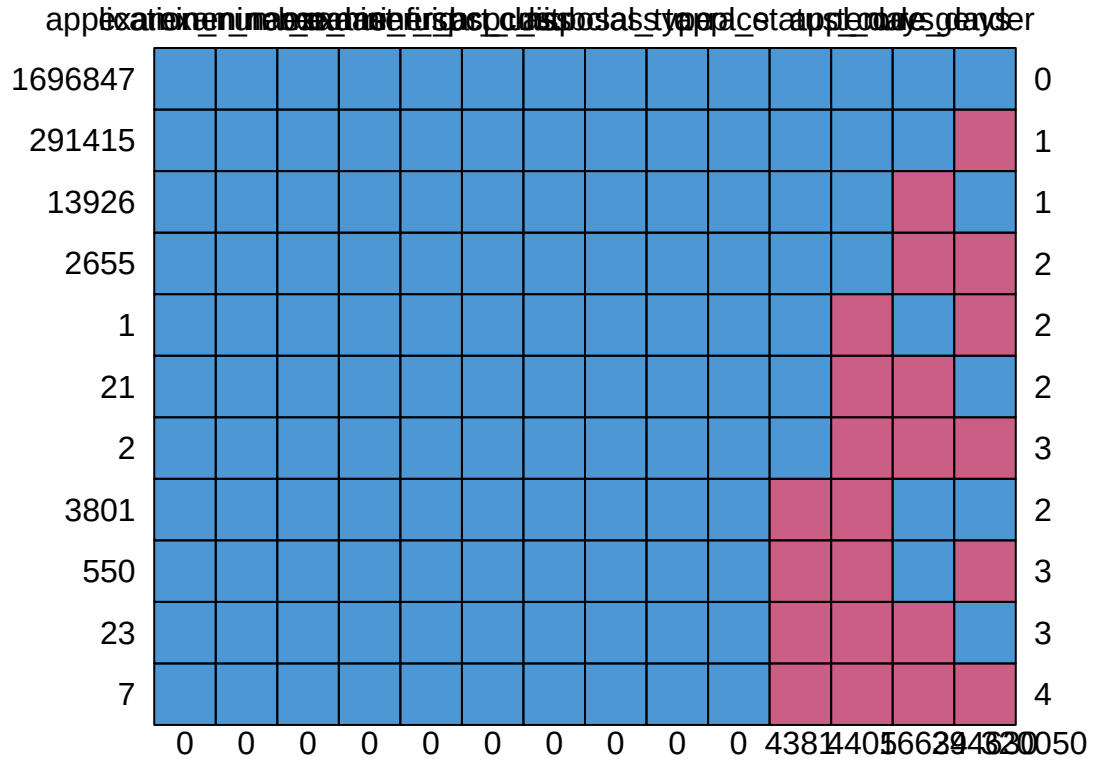
```
applications_subs = subset(applications, select=-c(examiner_name_middle,patent_number, appl_status_date
# Removal explanations:

# some people might not have a middle name by choice (ie it was not just randomly forgotten to be enter
# missing patent number means no patent issues, not missing at random
# appl_status_date for the same reason as patent number, and all of the related date-measurements arisi

# we remove the remaining date columns since we already have the metrics we need from them (tenure and

# we want examiner_id to remain unique which will not be the case if we allow mice to impute it, so we
applications_subs = applications_subs %>% drop_na(examiner_id)
```

```
library(mice)
md.pattern(applications_subs)
```




```
##         application_number examiner_name_last examiner_name_first examiner_id
## 1696847                  1                  1                    1           1
## 291415                   1                  1                    1           1
## 13926                    1                  1                    1           1
## 2655                     1                  1                    1           1
## 1                        1                  1                    1           1
## 21                       1                  1                    1           1
## 2                        1                  1                    1           1
## 3801                     1                  1                    1           1
## 550                      1                  1                    1           1
## 23                       1                  1                    1           1
## 7                        1                  1                    1           1
##                          0                  0                    0           0
##         examiner_art_unit uspc_class uspc_subclass disposal_type tc race
## 1696847                 1          1             1             1  1    1
## 291415                  1          1             1             1  1    1
## 13926                   1          1             1             1  1    1
## 2655                    1          1             1             1  1    1
## 1                       1          1             1             1  1    1
## 21                      1          1             1             1  1    1
## 2                       1          1             1             1  1    1
## 3801                    1          1             1             1  1    1
```

```
## 550                     1          1             1             1  1    1
## 23                      1          1             1             1  1    1
## 7                       1          1             1             1  1    1
##                         0          0             0             0  0    0
##         appl_status_code appl_days tenure_days gender
## 1696847                1         1           1      1      0
## 291415                 1         1           1      0      1
## 13926                  1         1           0      1      1
## 2655                   1         1           0      0      2
## 1                      1         0           1      0      2
## 21                     1         0           0      1      2
## 2                      1         0           0      0      3
## 3801                   0         0           1      1      2
## 550                    0         0           1      0      3
## 23                     0         0           0      1      3
## 7                      0         0           0      0      4
##                     4381      4405       16634 294630 320050
```

```
# there are 1696847 observations with no missing values (84% of the dataset)
# another 14% has just one missing value (gender)
# the remaining 2% of missing values is composed of the other features

applications_subs$gender = as.factor(applications_subs$gender) # mice will only impute on categorically

applications_full = complete(mice(applications_subs, m=3, maxit=3)) # impute using default mice imputat
```

```
##
##  iter imp variable
##    1   1  appl_status_code  gender  tenure_days  appl_days
##    1   2  appl_status_code  gender  tenure_days  appl_days
##    1   3  appl_status_code  gender  tenure_days  appl_days
##    2   1  appl_status_code  gender  tenure_days  appl_days
##    2   2  appl_status_code  gender  tenure_days  appl_days
##    2   3  appl_status_code  gender  tenure_days  appl_days
##    3   1  appl_status_code  gender  tenure_days  appl_days
##    3   2  appl_status_code  gender  tenure_days  appl_days
##    3   3  appl_status_code  gender  tenure_days  appl_days
```

```
rm(applications_subs)

applications_full %>% skim() # all done
```

Table 5: Data summary

Name	Piped data
Number of rows	2009248
Number of columns	14
Column type frequency:	
character	7

factor	1
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
application_number	0	1	8	8	0	2009248	0
examiner_name_last	0	1	2	17	0	3805	0
examiner_name_first	0	1	1	12	0	2594	0
uspc_class	0	1	3	3	0	414	0
uspc_subclass	0	1	6	6	0	6151	0
disposal_type	0	1	3	4	0	3	0
race	0	1	5	8	0	5	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	mal: 1347725, fem: 661523

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
examiner_id	0	1	78712.39	13606.61	59012	66476	75243	93754	99990	
examiner_art_unit	0	1	1928.59	304.48	1600	1671	1773	2171	2498	
appl_status_code	0	1	146.24	51.43	1	150	150	161	865	
tc	0	1	1877.47	298.92	1600	1600	1700	2100	2400	
tenure_days	0	1	5526.44	1101.26	27	4957	6078	6335	6518	
appl_days	0	1	1360.02	998.47	-160	741	1120	1665	6332	

With our remaining values imputed, we can proceed with constructing our advice network and calculating centralities

Advice networks

```
# first get work group for each examiner and limit to our two wgs of interest
examiner_aus = distinct(subset(applications_full, select=c(examiner_art_unit, examiner_id)))
# we eventually want to make a network with nodes colored by work group, so lets add that indicator
examiner_aus$wg = substr(examiner_aus$examiner_art_unit, 1,3)
# restrict down to our selected art units to reduce merging complexity later on
# examiner_aus = examiner_aus[examiner_aus$wg==163 | examiner_aus$wg==176,]

# now we will merge in the aus df on applications
adviceNet = merge(x=edges, y=examiner_aus, by.x="ego_examiner_id", by.y="examiner_id", all.x=TRUE)
```

```
adviceNet = adviceNet %>% rename(ego_art_unit=examiner_art_unit, ego_wg=wg)

# drop edges which are missing ego or alter id
adviceNet = drop_na(adviceNet)

# now repeat for the alter examiners
adviceNet = merge(x=adviceNet, y=examiner_aus, by.x="alter_examiner_id", by.y="examiner_id", all.x=TRUE)
adviceNet = adviceNet %>% rename(alter_art_unit=examiner_art_unit, alter_wg=wg)
adviceNet = drop_na(adviceNet)

egoNodes = subset(adviceNet, select=c(ego_examiner_id,ego_art_unit, ego_wg)) %>%   rename(examiner_id=eg
alterNodes = subset(adviceNet, select=c(alter_examiner_id,alter_art_unit, alter_wg))%>% rename(examiner_
nodes = rbind(egoNodes, alterNodes)
nodes = distinct(nodes) #5412 examiners(but some are repeated because they move amongst art units)

# when we reduce to the list of distinct vertices, we actually have more than we should, since some exa
nodes = nodes %>% group_by(examiner_id) %>% summarise(examiner_id=first(examiner_id), art_unit=first(ar
# we are left with just 2400 unique examiners
```

Construct network and calculate centralities

```
adviceNet = graph_from_data_frame(d=adviceNet, vertices=nodes, directed=TRUE)
# centralities
Degree <- degree(adviceNet, v=V(adviceNet))
Betweenness <- betweenness(adviceNet)
Eigenvector <- evcent(adviceNet)$vector

V(adviceNet)$size = Degree
V(adviceNet)$eig = round(Eigenvector,2)
V(adviceNet)$bet = round(Betweenness,2)
```

Model the relationship between centralities and app_proc_time

```
# first we'll need to merge the centrality measurements back into the imputed applications set
centralities <- cbind(Degree, Eigenvector, Betweenness)
centralities = round(centralities,2)
centralities = data.frame(centralities)
centralities <- cbind(examiner_id = rownames(centralities), centralities)
rownames(centralities) <- 1:nrow(centralities)

# now merge on examiner_id
applications_final = merge(x=applications_full, y=centralities, by="examiner_id", all.x=TRUE)
applications_final %>% skim() # we will have quite a few NaNs popping back up for those examiners who d
```

Table 9: Data summary

Name	Piped data
Number of rows	2009248
Number of columns	17
Column type frequency:	
character	7
factor	1
numeric	9
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
application_number	0	1	8	8	0	2009248	0
examiner_name_last	0	1	2	17	0	3805	0
examiner_name_first	0	1	1	12	0	2594	0
uspc_class	0	1	3	3	0	414	0
uspc_subclass	0	1	6	6	0	6151	0
disposal_type	0	1	3	4	0	3	0
race	0	1	5	8	0	5	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	mal: 1347725, fem: 661523

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
examiner_id	0	1.0	78712.39	13606.61	59012	66476	75243	93754	99990.00	
examiner_art_unit	0	1.0	1928.59	304.48	1600	1671	1773	2171	2498.00	
appl_status_code	0	1.0	146.24	51.43	1	150	150	161	865.00	
tc	0	1.0	1877.47	298.92	1600	1600	1700	2100	2400.00	
tenure_days	0	1.0	5526.44	1101.26	27	4957	6078	6335	6518.00	
appl_days	0	1.0	1360.02	998.47	-160	741	1120	1665	6332.00	
Degree	812768	0.6	99.72	181.65	1	12	35	102	2844.00	
Eigenvector	812768	0.6	0.00	0.02	0	0	0	0	1.00	
Betweenness	812768	0.6	308.71	2528.32	0	0	0	1	62882.35	

```
# nothing to do there but remove the missing values
applications_final = drop_na(applications_final)

# clean
rm(examiner_aus)
```

```
rm(egoNodes)
rm(alterNodes)
rm(nodes)
rm(edges)
rm(adviceNet)
gc()
```

```
##             used  (Mb) gc trigger   (Mb)  max used (Mb)
## Ncells   5165554 275.9   16383834  875.0  16383834  875
## Vcells 120073247 916.1  336903934 2570.4 421129917 3213
```

Modelling

```
# we wish to model the relationship between various centralities and appl_days
# we will make our first model as a simplistic model assuming no interactions among predictors
lm1 = lm(appl_days~Degree+Eigenvector+Betweenness+tenure_days+gender, data=applications_final)
summary(lm1)
```

```
##
## Call:
## lm(formula = appl_days ~ Degree + Eigenvector + Betweenness +
##     tenure_days + gender, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1682.6  -649.2  -239.5   343.2  5046.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.971e+02  7.653e+00  78.024  < 2e-16 ***
## Degree      -6.583e-02  5.411e-03 -12.167  < 2e-16 ***
## Eigenvector  1.336e+02  3.853e+01   3.467 0.000527 ***
## Betweenness  8.017e-03  3.782e-04  21.198  < 2e-16 ***
## tenure_days  1.347e-01  1.254e-03 107.392  < 2e-16 ***
## gendermale   2.107e+01  2.028e+00  10.389  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1018 on 1196474 degrees of freedom
## Multiple R-squared:  0.009993,   Adjusted R-squared:  0.009988
## F-statistic:  2415 on 5 and 1196474 DF,  p-value: < 2.2e-16
```

Interpretations: - The "baseline" expectation for application processing time is a little under 600 days - That would be for a female examiner who just started, 0 tenure days, and has never asked any advice

- Everytime an examiner asks advice to a new colleague examiner (increase degree by 1), we expect processing time to decrease slightly (.06 days)
- Increasing an examiner's importance as measured by eigenvector centrality is expected to increase processing time by 139 days

- Similarly, increasing an examiner's betweenness increases the processing time slightly (less than a day)
- It is important to note that the centrality measurements are all coupled, so in a vacuum these interpretations are valid, but in practice we could not increase an examiner's degree without also altering in some way their eigenvector and betweenness centralities
- Longer tenured examiners take slightly longer to process applications with each additional day of tenure
- Male examiners are expected to take roughly 22 days longer to process than women

We can try to capture some of the more complex relationships among predictors by adding interactions

```
lm2 = lm(appl_days~Degree+Eigenvector+Betweenness+tenure_days+gender+Degree*gender+Eigenvector*gender+B
summary(lm2)
```

```
##
## Call:
## lm(formula = appl_days ~ Degree + Eigenvector + Betweenness +
##     tenure_days + gender + Degree * gender + Eigenvector * gender +
##     Betweenness * gender, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1711.7  -650.5  -239.9   343.3  5041.4
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.225e+02  7.708e+00  80.758  < 2e-16 ***
## Degree                  -2.560e-01  8.931e-03 -28.666  < 2e-16 ***
## Eigenvector              2.200e+03  3.616e+02   6.085 1.17e-09 ***
## Betweenness              6.464e-03  9.470e-04   6.825 8.77e-12 ***
## tenure_days              1.334e-01  1.255e-03 106.346  < 2e-16 ***
## gendermale              -7.710e+00  2.293e+00  -3.362 0.000773 ***
## Degree:gendermale        2.998e-01  1.122e-02  26.723  < 2e-16 ***
## Eigenvector:gendermale -2.261e+03  3.637e+02  -6.218 5.04e-10 ***
## Betweenness:gendermale  7.002e-04  1.034e-03   0.677 0.498204
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1018 on 1196471 degrees of freedom
## Multiple R-squared:  0.01062,    Adjusted R-squared:  0.01062
## F-statistic:  1606 on 8 and 1196471 DF,  p-value: < 2.2e-16
```

- The baseline expectation has increased to 621 days in this new model
- While some values have changed, the directionality of each individual predictor is unaffected in this new model, except for gender. This new model now predicts male examiners to take 7 fewer days to process applications compared to female examiners
- From interaction terms, we also know that increasing degree for male examiners increases expected processing time by under a day
- Male examiners with higher eigenvector centrality have a significantly reduced processing time (226 days)

- Male examiners with higher betweenness centrality have roughly the same expected processing time (0.0003 days added)
- This would seem to imply that "importance" as measured by eigenvector centrality is more meaningful for male examiners than it is for female examiners

Lets investigate that insight with some predictions:

```
baseline = predict(lm2, data.frame(Degree=0,Eigenvector=0,Betweenness=0,tenure_days=0,gender='female'))
lowEigMale = predict(lm2, data.frame(Degree=0,Eigenvector=0,Betweenness=0,tenure_days=0,gender='male'))
lowEigFemale = baseline
highEigMale = predict(lm2, data.frame(Degree=0,Eigenvector=1,Betweenness=0,tenure_days=0,gender='male'))
highEigFemale = predict(lm2, data.frame(Degree=0,Eigenvector=1,Betweenness=0,tenure_days=0,gender='femal

data.frame(baseline=baseline, unimportant_male=lowEigMale, important_male=highEigMale, unimportant_femal
```

```
##   baseline unimportant_male important_male unimportant_female important_female
## 1 622.5221         614.8117        553.452           622.5221         2822.521
```

This verifies our insight from earlier: Importance seems to mean more for men than for women. We can't deduce why that is from this model, but conjecture might say that since this is a male-dominated organization, men seem to benefit (at least in terms of reducing processing time) from advice-seeking more than women.

Disclaimer: While these models are providing theoretically meaningful insights, we should note that the proportion of variance in the data explained by both of these models is around 1%, ie they are not particularly good models as far as goodness-of-fit is concerned.