Assignment: Staffing Database

Dunja Novaković

2020-07-21

Instructions

This assignment reviews the *Staffing Database* analytical lecture. You will use the *staffing_database.Rmd* file I reviewed in the video lectures to complete this assignment. You will *copy and paste* relevant code from that file and update it to answer the questions in this assignment. You will respond to questions in each section after executing relevant code to answer a question. You will submit this assignment to its *Submissions* folder on *D2L*. You will submit this *(1)* completed **R Markdown** script and *(2)* a *HTML* or *PDF* rendered version of it to *D2L* by the due date and time. If you installed `TinyTeX` successfully, then I prefer a *PDF* version.

To start:

For any analytical project, you want to create a clear project directory structure.
All materials from this course should exist in one folder on your computer. Inside of that main course folder, you should create folders to store course documentation, lecture analytical projects, assignments analytical projects, etc. Inside of your folder for assignments analytical projects, you should create folder for this assignment named *staffing_database*.

Any analytical project folder should contain inside it at least three additional folders named *scripts*, *data*, and *plots*. Store this script in the *scripts* folder, the data for this assignment in the *data* folder, and any requested plots in the *plots* folder. Each analytical project should also contain a **.Rproj** file in its top-level directory. Go to the *File* menu in *RStudio*, select *New Project…*, choose *Existing Directory*, go to the folder you created to contain this analytical project. Select it as the top-level directory for this **RStudio Project**.

Global Settings

The first code chunk sets the global settings for the remaining code chunks in the document. Do *not* change anything in this code chunk.

Load Packages

In this code chunk, we load three packages we need for this assignment:

1. **here**,
2. **tidyverse**,
3. **DBI**,
4. **RSQLite**,
5. **skimr**,
6. **GGally**,
7. **qgraph**, and
8. **plotly**.

We will use functions from these packages to import the data, examine the data, calculate summaries on the data, and create visualizations from the data. Do *not* change anything in this code chunk.

```
### load libaries for use in current working session
## here for workflow
library(here)

## tidyverse for data manipulation and plotting
# loads eight different libraries simultaneously
library(tidyverse)

## DBI to work with database
library(DBI)

## RSQLite to import database
library(RSQLite)

## skimr for summary statistics
library(skimr)

## GGally for plotting
library(GGally)

## qgraph for network plots
library(qgraph)
```

Task 1: Load Data

As your first task for this assignment, you need to load the data of interest. We will use the same database as in the analytical lecture: **staffing_database.sqlite**.

Use the appropriate functions to navigate to your *data* directory and import the database. Import the database as the object **staff_db**. Note the difference in the name of **staff_db** from the lecture script. List all of the data tables in **staff_db**.

Question 1.1: How many data tables are there in the database?

Response 1.1: *7*.

Use *SQL* to query the data table named *cv* in the database and print the first *8* rows.

Question 1.2: Is the employee with *id = 232* a minority? Does this same employee have prior work experience?

Response 1.2: *The employee in question is not a minority and doesn't have prior work experience.*

Save each of the data tables in the database as a *tibble* data object. Use the same names as in the lecture script. Disconnect from the database. Use **arrange** on **onboard_data** to sort the data by **id** such that the person with **id** equal to one is listed first. Make sure you print the data.

Question 1.3: Out of the first 10 individuals (i.e., individuals with **id** from one to ten),how many did *not* get an *onboarding buddy*?

Response 1.3: *5*.

```
#### Q1.1
### import database
```

```r
## use here() to locate file in our project directory;
## use DBI::dbConnect to open connection;
## RSQLite::SQLite to import this particular database
staff_db <- dbConnect(SQLite(), here("data", "staffing_database.sqlite"))
### list all of the data tables
dbListTables(staff_db)
```

```
## [1] "ac"       "cv"       "ids"      "jobs"     "managers" "onboard"  "outcomes"
```

```r
#### Q1.2
### extract information from a table with SQL code
dbGetQuery(staff_db, "SELECT * FROM cv LIMIT 8")
```

```
##    id gender minority education work_exp
## 1  47   Male       No       BSc       No
## 2 227   Male       No       BSc       No
## 3 229   Male       No       MSc      Yes
## 4 231   Male       No       BSc      Yes
## 5 232   Male       No       BSc       No
## 6   7   Male      Yes       BSc      Yes
## 7   8   Male      Yes       BSc       No
## 8   9   Male      Yes       BSc      Yes
```

```r
#### Q1.3
### save database table to tibble object
## ac
ac_data <- tbl(staff_db, "ac") %>% as_tibble()
## cv
cv_data <- tbl(staff_db, "cv") %>% as_tibble()
## ids
ids_data <- tbl(staff_db, "ids") %>% as_tibble()
## jobs
jobs_data <- tbl(staff_db, "jobs") %>% as_tibble()
## managers
managers_data <- tbl(staff_db, "managers") %>% as_tibble()
## onboard
onboard_data <- tbl(staff_db, "onboard") %>% as_tibble()
## outcomes
outcomes_data <- tbl(staff_db, "outcomes") %>% as_tibble()

### disconnect from database
dbDisconnect(staff_db)

### arranging data
## choose data
onboard_data %>%
  ## arrange by id
  arrange(id)
```

```
## # A tibble: 360 x 4
##       id InductionDay InductionWeek OnBoardingBuddy
##    <dbl>        <dbl>         <dbl>           <dbl>
```

```
## 1     1            1             1              1
## 2     2            1             0              0
## 3     3            1             0              1
## 4     4            1             0              0
## 5     5            1             0              0
## 6     6            0             0              0
## 7     7            1             1              1
## 8     8            1             1              1
## 9     9            1             1              1
## 10   10            1             0              0
## # ... with 350 more rows
```

Task 2: Joins

For the second task, you will join the various data tables into one complete data object named **staff_join**. Start by joining the following five data tables in one chained (i.e., use the pipe operator to link the joins together) command:

1. **ids_data**,
2. **cv_data**,
3. **ac_data**,
4. **onboard_data**, and
5. **outcomes_data**.

Question 2.1: After joining these five tables, how many variables are in **staff_join**?

Response 2.1: *19.*

Next, join **managers_data** and **jobs_data** to **staff_join**. Rename the **span** and **budget** variables to **mgr_span** and **job_budget** as in the lecture script.

Question 2.2: After joining these two tables, how many variables are in **staff_join**?

Response 2.2: *21.*

Mutate the variables in **staff_join** as in the lecture script. Use **glimpse** on **staff_join** after completing the mutations.

Question 2.3: How many total nominal (i.e., *fct*) and ordered (i.e., *ord*) factor variables are there in **staff_join** after the mutations?

Response 2.3: *Nominal: 7; Ordered: 2*

```
#### Q2.1
### join tables
staff_join <- ids_data %>%
  ## join ids with cv
  left_join(cv_data, by = c("emp_id" = "id"))%>%
  ## join ac_data
  left_join(ac_data, by = c("emp_id" = "id")) %>%
  ## join ac_data
  left_join(onboard_data, by = c("emp_id" = "id")) %>%
  ## join ac_data
  left_join(outcomes_data, by = c("emp_id" = "id"))

#### Q2.2
```

```r
### overwrite current joined data
staff_join <- staff_join %>%
  # join managers
  left_join(managers_data, by = c("mgr_id" = "id")) %>%
  # join jobs
  left_join(jobs_data, by = c("job" = "unit")) %>%
  # rename joined variables
  rename(mgr_span = span, job_budget = budget)

#### Q2.3
### manipulate character variables to factors
## overwrite data
staff_join <- staff_join %>%
  ## select nominal factors
  mutate_at(vars(job:work_exp, -education,
                 InductionDay:OnBoardingBuddy), as_factor) %>%
  ## select ordered factor
  mutate_at(vars(education, left), factor, ordered = TRUE) %>%
  ## recode onboarding factors
  mutate_at(vars(InductionDay:OnBoardingBuddy),
            ~fct_recode(., `No` = "0", `Yes` = "1")) %>%
  ## recode factor
  mutate_at(vars(left),
            ~fct_recode(., `No` = "0", `One` = "1", `Two` = "2")) %>%
  ## relevel factor
  mutate_at(vars(left), ~fct_relevel(., "No", after = 2))

### using glimpse
glimpse(staff_join)
```

```
## Rows: 360
## Columns: 21
## $ emp_id          <dbl> 47, 227, 229, 231, 232, 7, 8, 9, 10, 43, 44, 45, 58, 5~
## $ mgr_id          <int> 1, 8, 3, 8, 2, 9, 9, 16, 12, 9, 14, 13, 13, 16, 14, 15~
## $ job             <fct> HR, HR, HR, HR, HR, Finance, Finance, Finance, Finance~
## $ gender          <fct> Male, Male, Male, Male, Male, Male, Male, Male, Male, ~
## $ minority        <fct> No, No, No, No, No, Yes, Yes, Yes, No, No, No, No, No,~
## $ education       <ord> BSc, BSc, MSc, BSc, BSc, BSc, BSc, BSc, BSc, MSc, MSc,~
## $ work_exp        <fct> No, No, Yes, Yes, No, Yes, No, Yes, No, Yes, Yes, No, ~
## $ open            <dbl> 46, 46, 46, 57, 35, 35, 36, 26, 57, 77, 57, 46, 36, 68~
## $ consc           <dbl> 57, 57, 57, 45, 68, 78, 89, 78, 89, 68, 46, 78, 46, 80~
## $ extra           <dbl> 35, 35, 26, 45, 57, 35, 36, 26, 57, 77, 57, 46, 57, 68~
## $ agree           <dbl> 87, 87, 89, 78, 98, 35, 36, 26, 57, 77, 57, 46, 35, 68~
## $ neuro           <dbl> 26, 26, 35, 36, 45, 67, 89, 89, 97, 89, 89, 89, 35, 45~
## $ cog_quant       <dbl> 56, 56, 68, 79, 97, 89, 90, 89, 90, 97, 96, 89, 92, 97~
## $ cog_verb        <dbl> 87, 87, 89, 88, 78, 84, 78, 83, 82, 78, 78, 83, 83, 79~
## $ InductionDay    <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes,~
## $ InductionWeek   <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, No, Yes, Yes, ~
## $ OnBoardingBuddy <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, No, Yes, Yes, ~
## $ perf            <dbl> 34, 48, 44, 48, 41, 68, 73, 78, 61, 32, 32, 33, 48, 38~
## $ left            <ord> No, No, No, No, No, No, No, No, No, No, No, No, No, No~
## $ mgr_span        <int> 6, 13, 10, 13, 8, 7, 7, 11, 11, 7, 7, 10, 10, 11, 7, 9~
## $ job_budget      <dbl> 3.2, 3.2, 3.2, 3.2, 3.2, 4.5, 4.5, 4.5, 4.5, 4.5, 4.5,~
```

Task 3: Data Transformations

For your third task, you will transform **staff_join** to answer questions.

Select **emp_id**, **cog_quant**, and **open** from **staff_join**. Arrange by ascending **cog_quant** and descending **open**.

Question 3.1: What employee (i.e., **emp_id**) is listed *first*? What is the **open** score for employee with **emp_id** equal to *22*?

Response 3.1: *The id of the employee that is listed first is 47. The open score for the employee with emp_id=22 is 57.*

Select **emp_id**, **cog_verb**, **neuro**, and **consc** from **staff_join**. Filter for the *top 15%* of employees on **neuro**. Arrange by descending **cog_verb** and ascending **consc**.

Question 3.2: What employee (i.e., **emp_id**) is listed *fifth*? What is the **consc** score for employee with **emp_id** equal to *19*?

Response 3.2: *The id of the employee that is listed fifth is 16. The consc score for the employee with emp_id=19 is 46.*

Select **emp_id**, **education**, and **agree** from **staff_join**. Filter for indvididuals with a **PhD education** and **agree** scores greater than *88*.

Question 3.3: Which two employees (i.e., **emp_id**) meet the criteria?

Response 3.3: *Two employees with emp_ids equal to 33 and 213.*

Select **OnBoardingBuddy**, **gender**, **agree**, and **consc** from **staff_join**. Group by **OnBoardingBuddy** and **gender**. Compute the *minimum*, *median*, and *max* for each group. Pay attention to appropriately using *commas* and *parentheses*. Remove the groups with **ungroup()**. Pivot the table longer via **pivot_longer()**. Adjust the **cols** input correctly inside of **pivot_longer()**. Print all rows using **print()** setting the **n** input correctly.

Question 3.4: What is the median *agreeableness* score for males who had an onboarding buddy? What is the minimum *conscientiousness* score for females who did *not* have an onboarding buddy?

Response 3.4: *46; 26.*

```r
#### Q3.1
###aranging data
## choose data
staff_join %>%
  ## select variables
  select(emp_id, cog_quant, open) %>%
  ## arrange
  arrange(cog_quant, desc(open))
```

```
## # A tibble: 360 x 3
##    emp_id cog_quant  open
##     <dbl>     <dbl> <dbl>
##  1     47        56    46
##  2    227        56    46
##  3     28        56    45
##  4    208        56    45
##  5      6        66    45
##  6    186        66    45
##  7     22        67    57
##  8    202        67    57
```

```
## 9     29         67    35
## 10    209        67    35
## # ... with 350 more rows
```

```r
#### Q3.2
### arranging data
## choose data
staff_join %>%
  ## select variables
  select(emp_id, cog_verb, neuro, consc) %>%
  ## top neuro scores
  top_frac(0.15, neuro) %>%
  ## arrange
  arrange(desc(cog_verb), consc)
```

```
## # A tibble: 64 x 4
##    emp_id cog_verb neuro consc
##     <dbl>    <dbl> <dbl> <dbl>
##  1      4       97    97    26
##  2    184       97    97    26
##  3      3       97    97    87
##  4    183       97    97    87
##  5     16       97    89    98
##  6    196       97    89    98
##  7     17       96    89    89
##  8    197       96    89    89
##  9     19       92    98    46
## 10    199       92    98    46
## # ... with 54 more rows
```

```r
#### Q3.3
### filtering data
## choose data
staff_join %>%
  ## select variables
  select(emp_id, education, agree) %>%
  ## filter for PhD education;
  ## AND agree greater than 88
  filter(education == "PhD", agree > 88)
```

```
## # A tibble: 2 x 3
##   emp_id education agree
##    <dbl> <ord>     <dbl>
## 1     33 PhD          89
## 2    213 PhD          89
```

```r
#### Q3.4
### summarizing data
## choose data
staff_join %>%
  ## select variables
  select(OnBoardingBuddy, gender, agree, consc) %>%
```

```r
## group by variable
group_by(OnBoardingBuddy, gender) %>%
## summarize
summarize_all(list(~min(., na.rm = T),
                   ~median(., na.rm = T),
                   ~max(., na.rm = T))) %>%
## remove grouping
ungroup() %>%
## pivot longer
             # choose columns to make longer
pivot_longer(cols = agree_min:consc_max,
             # new column for names of variables
             names_to = c("var", "stat"),
             # create new columns by separator
             names_sep = "_",
             # new column for values of variables
             values_to = "value") %>%
## print all rows
print(n = Inf)
```

```
## # A tibble: 24 x 5
##    OnBoardingBuddy gender var   stat   value
##    <fct>           <fct>  <chr> <chr>  <dbl>
##  1 No              Male   agree min       35
##  2 No              Male   consc min       79
##  3 No              Male   agree median    57
##  4 No              Male   consc median    80
##  5 No              Male   agree max       98
##  6 No              Male   consc max       89
##  7 No              Female agree min       26
##  8 No              Female consc min       26
##  9 No              Female agree median    57
## 10 No              Female consc median    57
## 11 No              Female agree max       97
## 12 No              Female consc max       98
## 13 Yes             Male   agree min       26
## 14 Yes             Male   consc min       26
## 15 Yes             Male   agree median    46
## 16 Yes             Male   consc median    68
## 17 Yes             Male   agree max       98
## 18 Yes             Male   consc max       98
## 19 Yes             Female agree min       26
## 20 Yes             Female consc min       23
## 21 Yes             Female agree median    57
## 22 Yes             Female consc median    57
## 23 Yes             Female agree max       98
## 24 Yes             Female consc max       98
```

Task 4: Descriptive Summaries

For this task, you will compute descriptive summaries on **staff_join**.

Select **education**, **minority**, **gender**, and all *5* personality variables (i.e., **open**, **consc**, **extra**, **agree**, and

neuro) from **staff_join**. Group by **education**. Use **skim_without_charts()** to compute summaries for the groups.

Question 4.1: How many Master's (i.e., **MSc**) educated employees are *minorities* (i.e., **minority**) in the company? What is the average *extraversion* (i.e., **extra**) score employees with a *PhD*?

Response 4.1: *40; 65.*

Compute the correlations between **cog_quant**, **cog_verb**, and **perf**.

Question 4.2: What is the correlation between **cog_verb** and **perf**?

Response 4.2: *0.3790663.*

Save the following as the object named **dist_vars**: First, filter by **job** equals to **Risk** and **minority** equals to **Yes**. Second, select all *5* personality variables (i.e., **open**, **consc**, **extra**, **agree**, and **neuro**), **cog_quant**, **cog_verb**, and **perf** from **staff_join**. Third, compute the *distance* between selected individuals. Fourth, use **round()** to round the distances to two digits. Fifth, convert the object to a matrix.

Question 4.3: What is the computed distance between the third and sixth individual? Which two indivdiuals are most similar (i.e., least distant, lowest distance score)?

Response 4.3: *The computed distance between the third and sixth individual is 80.19. The most similar individuals are the first and second individual (if we exclude distance scores on the main diagonal of the distance matrix).*

```
#### Q4.1
### compute summary statistics
## filtered group data
# choose data
staff_join %>%
  #select variables
  select(education, minority, gender, open:neuro)%>%
  # grouping variable
  group_by(education) %>%
  # summary
  skim_without_charts()
```

Table 1: Data summary

Name	Piped data
Number of rows	360
Number of columns	8
Column type frequency:	
factor	2
numeric	5
Group variables	education

Variable type: factor

skim_variable	education	n_missing	complete_rate	ordered	n_unique	top_counts
minority	BSc	0	1	FALSE	2	No: 104, Yes: 40

skim_variable	education	n_missing	complete_rate	ordered	n_unique	top_counts
minority	MSc	0	1	FALSE	2	No: 166, Yes: 40
minority	PhD	0	1	FALSE	1	No: 10, Yes: 0
gender	BSc	0	1	FALSE	2	Mal: 84, Fem: 60
gender	MSc	0	1	FALSE	2	Mal: 120, Fem: 86
gender	PhD	0	1	FALSE	2	Mal: 6, Fem: 4

Variable type: numeric

skim_variable	education	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
open	BSc	0	1	43.26	13.62	26	35.00	45.0	57.00	97
open	MSc	0	1	52.90	14.27	26	45.00	57.0	57.00	97
open	PhD	0	1	62.40	22.20	46	46.00	46.0	77.00	97
consc	BSc	0	1	61.19	23.70	24	36.00	57.0	89.00	98
consc	MSc	0	1	62.90	19.17	23	46.00	58.0	79.75	98
consc	PhD	0	1	68.80	20.06	46	46.00	77.0	86.00	89
extra	BSc	0	1	55.33	21.74	26	36.00	51.5	76.50	98
extra	MSc	0	1	58.35	20.62	26	45.00	57.0	77.00	98
extra	PhD	0	1	65.00	20.25	46	46.00	57.0	87.00	89
agree	BSc	0	1	50.32	21.87	26	35.00	45.0	57.00	98
agree	MSc	0	1	61.08	20.64	26	45.00	57.0	78.00	98
agree	PhD	0	1	66.80	23.33	35	46.00	77.0	87.00	89
neuro	BSc	0	1	64.36	23.31	12	45.75	67.0	87.00	98
neuro	MSc	0	1	55.38	20.61	15	36.00	57.0	68.00	97
neuro	PhD	0	1	56.60	19.81	35	36.00	57.0	77.00	78

```r
#### Q4.2
### compute correlations
## choose data
staff_join %>%
  ## select variables
  select(cog_quant, cog_verb, perf)%>%
  ## compute correlations
  cor(use = "pairwise")
```

```
##           cog_quant  cog_verb      perf
## cog_quant 1.0000000 0.1411270 0.3120981
## cog_verb  0.1411270 1.0000000 0.3790663
## perf      0.3120981 0.3790663 1.0000000
```

```r
#### Q4.3
### compute distances
## choose data
dist_vars <- staff_join %>%
  ## filter for Risk with minority employees
  filter(job == "Risk", minority == "Yes") %>%
  ## select variables
  select(open:neuro, cog_quant, cog_verb, perf) %>%
  ## compute distances
  dist() %>%
```

```
  ## round numbers
  round(digits = 2) %>%
  ##convert to matrix
  as.matrix()
### print data
dist_vars
```

```
##       1     2     3      4      5     6
## 1  0.00 21.17 43.05  38.41  74.37 67.52
## 2 21.17  0.00 47.93  32.02  88.56 85.28
## 3 43.05 47.93  0.00  77.92  58.07 80.19
## 4 38.41 32.02 77.92   0.00 108.56 90.56
## 5 74.37 88.56 58.07 108.56   0.00 51.79
## 6 67.52 85.28 80.19  90.56  51.79  0.00
```

Task 5: Data Visualization

For this task, you will visualize the data from **staff_join**.

You will make a heatmap using **dist_vars**. First, keep only the upper triangle of values in **dist_vars** using the code from the lecture. Second, overwrite **dist_vars** to make it a long table instead of square matrix using the code from the lecture. Third, produce a heatmap named **heatmap_ggplot** adjusting the heatmap scale so the midpoint is *65* and the maximum is *130*. Otherwise, keep the code the same as in the lecture. Print the plot. Save the plot to your **plots** folder as **heatmap.png**.

Question 5.1: Looking at the plot, which two individuals are most distant (i.e., look for the bluest tile) on these variables?

Response 5.1: *The most distant individuals are the fourth and fifth one.*

Select **minority**, **education**, **consc**, **cog_quant**, and **perf** from **staff_join**. Use **ggpairs()** to produce a scatterplot matrix.

Question 5.2: What is the correlation between **consc** and **cog_quant**?

Response 5.2: *0.344*.

Compute a new object named **group_means** with the same code from the lecture but change the **skim_variable** to equal **neuro**. Next, compute a new object named **dist_means** with the same code from the lecture without any changes. Name the rows and columns of **dist_means** with the same code from the lecture. Apply **qgraph()** to **dist_means** just like the code from the lecture.

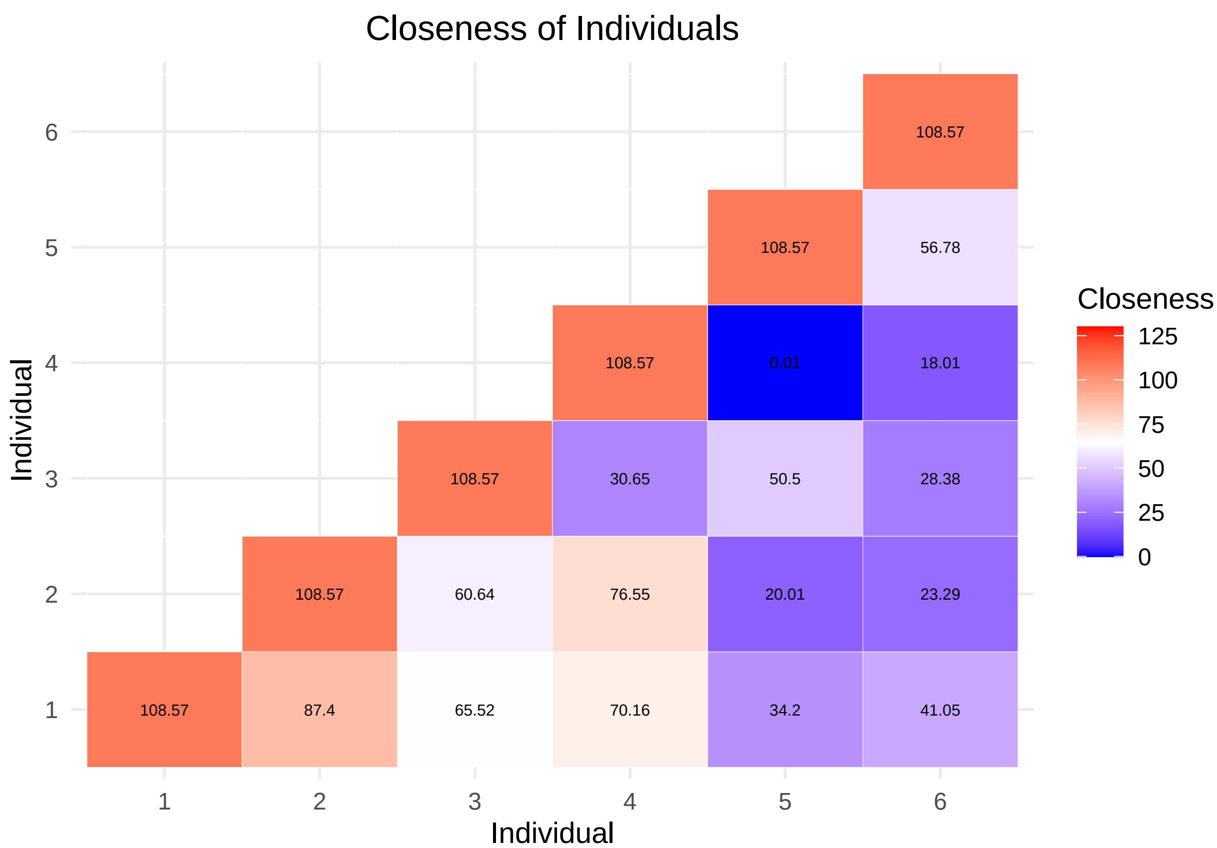
Question 5.3: Looking at the plot, which two jobs have the highest mean difference (i.e., thickest green line) on **neuro**?

Response 5.3: *Finance and Sales.*

```
#### Q5.1
### keep only upper triangle
## overwrite data
dist_vars[lower.tri(dist_vars)] <- NA

### pivot data
## overwrite data
dist_vars <- dist_vars %>%
  ## convert to tibble
  as_tibble() %>%
```

```r
  ## add row names
  rownames_to_column("Ind_1") %>%
  ## pivot longer
  pivot_longer(cols = -Ind_1, names_to = "Ind_2", values_to = "value") %>%
  ## mutate value
  mutate(value = max(value, na.rm = TRUE) - value + 0.01) %>%
  ## mutate character to factor
  mutate_if(is_character, as_factor)

### make plot
## set data and mapping
heatmap_ggplot <- ggplot(dist_vars, aes(x = Ind_2, y = Ind_1, fill = value)) +
  ## tile geometry
  geom_tile(color = "white") +
  ## color the tiles
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 65, limit = c(0, 130),
                       space = "Lab", name = "Closeness",
                       na.value = "transparent") +
  ## text geometry
  geom_text(aes(label = value), color = "black", size = 2) +
  ## white background
  theme_minimal() +
  ## axes labels
  labs(x = "Individual", y = "Individual") +
  ## title of plot
  ggtitle("Closeness of Individuals") +
  ## position of title
  theme(plot.title = element_text(hjust = 0.5))

## print heatmap
heatmap_ggplot
```

```
## Warning: Removed 15 rows containing missing values (geom_text).
```




```
## save heatmap
ggsave(here("plots", "heatmap.png"),
       # specify plot to save
       plot = heatmap_ggplot,
       # specify units
       units = "in", width = 9, height = 5)
```

```
## Warning: Removed 15 rows containing missing values (geom_text).
```

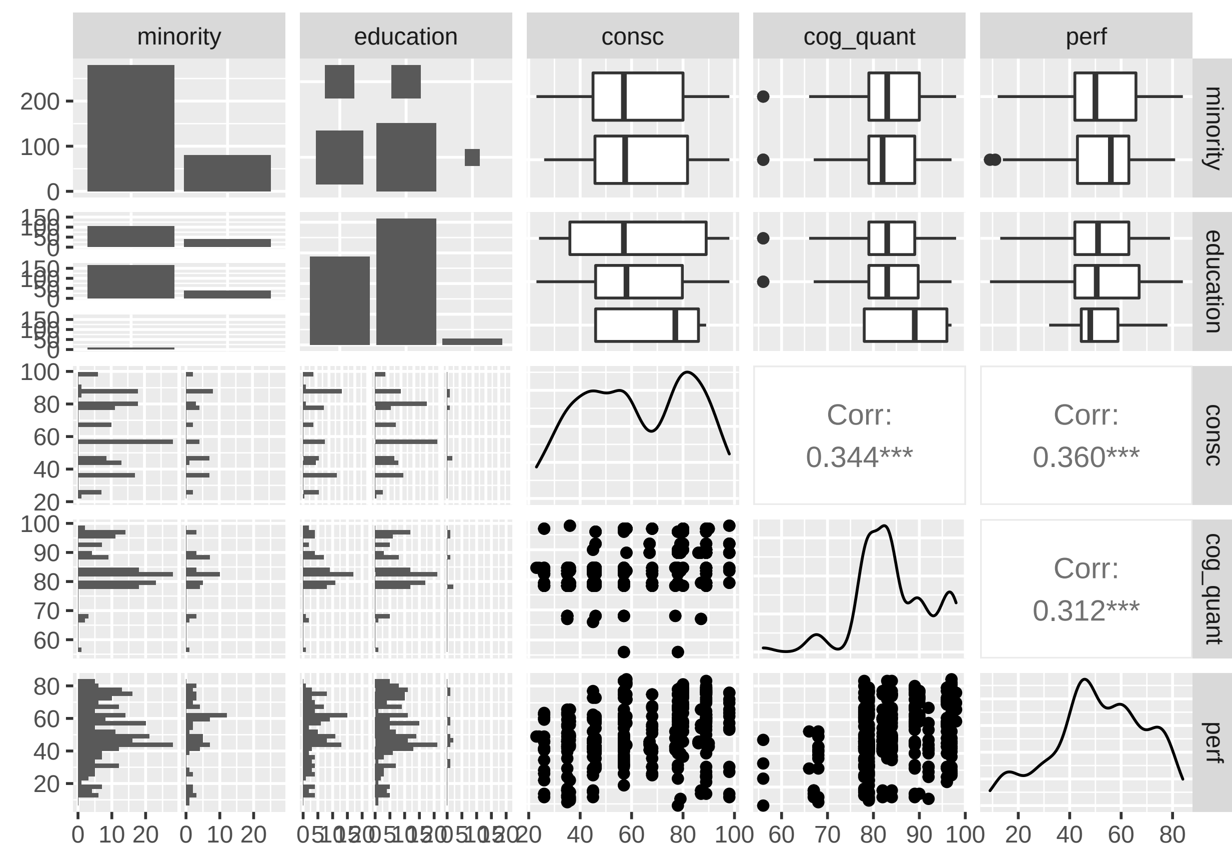
```
#### Q5.2
### choose data
staff_join %>%
  ## select variables
  select(minority, education, consc, cog_quant, perf) %>%
  ## scatterplot matrix
  ggpairs()
```

```
## Warning: Removed 15 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 15 rows containing non-finite values (stat_boxplot).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 15 rows containing missing values

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 15 rows containing missing values

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 15 rows containing non-finite values (stat_bin).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 15 rows containing non-finite values (stat_bin).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing non-finite values (stat_density).
```



```r
#### Q5.3
### distances between groups
## choose data
group_means <- staff_join %>%
  ## grouping variable
  group_by(job) %>%
  ## summary
  skim() %>%
  ## filter
  filter(skim_variable == "neuro") %>%
  ## select
  select(job, numeric.mean)
## compute distance matrix
dist_means <- group_means %>%
  ## select means variable
  select(numeric.mean) %>%
  ## compute distance
  dist(method = "manhattan") %>%
  ## convert to matrix
  as.matrix()
## name columns
colnames(dist_means) <- row.names(dist_means) <- group_means$job
## plot
qgraph(dist_means, layout = "spring")
```