Assignment: Mid-Term Review

Dunja Novaković

2020-07-30

Instructions

This assignment reviews the first four weeks of lectures and requires you to apply what you learned. You will use all available materials from the first four weeks of the course, including your previously completed assignments, to complete this *Mid-Term Review*.
Per usual, you will complete several tasks to answer questions. 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 a folder for this assignment named *mid_term_review*.

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.

Task 1: Load Packages

Unlike previous assignments, you will specify the packages to load for this *Mid-Term Review*. Load the following packages:

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

8. **broom**, and
9. **relaimpo**.

You will use functions from these packages to import the data, examine the data, calculate summaries on the data, build regression models, and create visualizations from the data.

Question 1.1: What does the message from the **here** package say?

Response 1.1: *here() starts at C:/Users/novak/OneDrive/Desktop/MGT 591/Assignments/mid_term_review.*

```
#### Q1.1
### load libaries for use in current working session
## here for workflow
library(here)
```

```
## here() starts at C:/Users/novak/OneDrive/Desktop/MGT 591/Assignments/mid_term_review
```

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## DBI to work with database
library(DBI)

## RSQLite to import database
library(RSQLite)

## skimr for summary statistics
library(skimr)

## GGally for plotting
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

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

## broom to work with model objects
```

```r
library(broom)

## relaimp to calculate relative predictor importance
library(relaimpo)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: boot

## Loading required package: survey

## Loading required package: grid

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart

## Loading required package: mitools

## This is the global version of package relaimpo.

## If you are a non-US user, a version with the interesting additional metric pmvd is available

## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

Task 2: Load Data

Use the appropriate functions to navigate to your *data* directory and import **org_db.sqlite**. Import the database as the object **org_db**. List all of the data tables in **org_db**.

Question 2.1: How many data tables are there in the database?

Response 2.1: *3*.

Use *SQL* to query the data table named **employees** in the database and print the first *12* rows.

Question 2.2: Is the employee with **id** = *E10060* older or younger (**age**) than *30*? What is the **compensation** of employee *E1008*? Does employee *E10012* or *E10086* have a higher **career_satisfaction** score?

Response 2.2: *The employee with id=E10060 is older than 30. The compensation of employee E1008 is 40380. Employee E10012 has a higher career_satisfaction score than employee E10086.*

Save each of the data tables in the database as a *tibble* data object. Name the *tibble* data objects exactly the same as the data tables of the database. Disconnect from the database.

Use **arrange** on **managers** to sort the data by *descending* **effectiveness** such that the managers with top **effectiveness** scores are listed first. Make sure the results prints to the console window.

Use **arrange** on **employees** to sort the data by *descending* **hiring_score** such that employees with top **hiring_score** values are listed first.

Question 2.3: Which three managers (**id**) have the highest **effectiveness** scores?
Which three employees (**id**) have the highest **hiring_score** values?

Response 2.3: *Managers: E8030, E5879, E8015. Employees: E7037, E6668, E3153.*

```
#### Q2.1
### import database
## use here() to locate file in our project directory;
## use DBI::dbConnect to open connection;
## RSQLite::SQLite to import this particular database
org_db <- dbConnect(SQLite(), here("data", "org_db.sqlite"))
### list all of the data tables
dbListTables(org_db)
```

```
## [1] "emp_mgr_ids" "employees"   "managers"
```

```
#### Q2.2
### extract information from a table with SQL code
dbGetQuery(org_db, "SELECT * FROM employees LIMIT 12")
```

```
##        id location level gender   age rating compensation percent_hike
## 1  E10012        1     1      1 25.09      4        64320           10
## 2  E10025        2     1      1 25.98      3        48204            8
## 3  E10027        3     2      1 33.40      3        85812           11
## 4  E10048        2     2      2 24.55      3        49536            8
## 5  E10060        3     1      2 31.23      3        75576           12
## 6  E10061        3     1      2 31.98      2        56904            8
## 7  E10065        2     1      2 24.84      3        38772           12
## 8  E10066        3     1      2 32.25      4        52320            9
## 9  E10078        1     1      1 29.02      3        50940            9
## 10  E1008        1     1      2 30.14      3        40380            6
```

```
## 11 E10083         2     1      1 27.13      4        57900            11
## 12 E10086         3     1      2 28.14      3        70152             7
##    hiring_score hiring_source no_previous_companies_worked distance_from_home
## 1            70             1                            0                 14
## 2            70             2                            9                 21
## 3            77             1                            3                 15
## 4            71             3                            5                  9
## 5            70             2                            0                 25
## 6            75             3                            8                 23
## 7            72             4                            9                 17
## 8            70             2                            6                 16
## 9            70             1                            1                 22
## 10           70             4                            3                 22
## 11           70             3                            3                 18
## 12           74             5                            6                 11
##    total_dependents marital_status education promotion_last_2_years
## 1                 2              1         1                      1
## 2                 2              1         1                      1
## 3                 5              1         1                      2
## 4                 3              1         1                      2
## 5                 4              1         1                      1
## 6                 5              1         2                      1
## 7                 2              1         1                      1
## 8                 5              1         1                      1
## 9                 2              1         1                      1
## 10                5              1         1                      1
## 11                5              1         1                      2
## 12                5              1         1                      1
##    no_leaves_taken total_experience monthly_overtime_hrs turnover
## 1                2             6.86                    1        0
## 2               10             4.88                    5        0
## 3               18             8.55                    3        0
## 4               19             4.76                    8        0
## 5               25             8.06                    1        0
## 6               15            13.72                    7        1
## 7               10             5.81                    2        0
## 8               20             7.56                   10        0
## 9               22             7.48                    2        0
## 10              23             8.40                   10        1
## 11              24             4.59                    8        0
## 12               2             6.00                    3        1
##    career_satisfaction perf_satisfaction work_satisfaction
## 1                 0.73              0.73              0.75
## 2                 0.72              0.84              0.85
## 3                 0.85              0.80              0.87
## 4                 0.42              0.33              0.85
## 5                 0.78              0.67              0.80
## 6                 0.88              0.81              0.86
## 7                 0.68              0.57              0.75
## 8                 0.76              0.74              0.95
## 9                 0.33              0.50              0.87
## 10                1.00              0.80              0.88
## 11                0.50              0.21              0.76
## 12                0.62              0.77              0.82
```

```r
#### Q1.3
### save database table to tibble object
## emp_mgr_ids
emp_mgr_ids <- tbl(org_db, "emp_mgr_ids") %>% as_tibble()
## employees
employees <- tbl(org_db, "employees") %>% as_tibble()
## managers
managers <- tbl(org_db, "managers") %>% as_tibble()

### disconnect from database
dbDisconnect(org_db)

### arranging data
## choose data
managers %>%
  ## arrange by id
  arrange(desc(effectiveness))
```

```
## # A tibble: 350 x 5
##    id     rating   age tenure effectiveness
##    <chr>   <dbl> <dbl>  <dbl>         <dbl>
##  1 E8030       3  25.9   3.84          1
##  2 E5879       3  34.2   0.58          1
##  3 E8015       4  33.2   3.32          1
##  4 E13918      5  35.1   2.25          0.99
##  5 E12078      4  30.3   1.23          0.99
##  6 E7471       3  42.6   1.62          0.99
##  7 E73         4  33.0  10.5           0.98
##  8 E77         4  27.0   6.58          0.98
##  9 E4788       3  29.3   0.24          0.98
## 10 E10422      4  33.3   2.73          0.98
## # ... with 340 more rows
```

```r
### arranging data
## choose data
employees %>%
  ## arrange by id
  arrange(desc(hiring_score))
```

```
## # A tibble: 1,954 x 23
##    id     location level gender   age rating compensation percent_hike
##    <chr>     <dbl> <dbl>  <dbl> <dbl>  <dbl>        <dbl>        <dbl>
##  1 E7037         3     1      2  28.8      4        90984           11
##  2 E6668         3     1      2  40.3      4        89100           15
##  3 E3153         3     1      2  39.0      4        75780           12
##  4 E2453         3     1      2  24.8      4        42384           13
##  5 E10681        3     2      1  33.7      3        70740            8
##  6 E11184        3     2      2  34.2      3        97236           10
##  7 E12139        3     1      2  25.3      4        42204           15
##  8 E12495        3     1      2  35        3        75192           14
##  9 E3448         3     1      2  34.7      2        62376            8
## 10 E5473         3     1      2  29.6      3        41280            9
## # ... with 1,944 more rows, and 15 more variables: hiring_score <dbl>,
```

```
## #   hiring_source <dbl>, no_previous_companies_worked <dbl>,
## #   distance_from_home <dbl>, total_dependents <dbl>, marital_status <dbl>,
## #   education <dbl>, promotion_last_2_years <dbl>, no_leaves_taken <dbl>,
## #   total_experience <dbl>, monthly_overtime_hrs <dbl>, turnover <dbl>,
## #   career_satisfaction <dbl>, perf_satisfaction <dbl>, work_satisfaction <dbl>
```

Task 3: Join and Clean Data

Join the individual data tables into one complete data object named **org_data**. First, join **emp_mgr_ids** with **employees**. Second, join the result of the previous step with **managers**. You will need to add the **suffix** argument inside the **left_join()** in the second step to adjust the names of variables with common names in **employees** and **managers**. Use the argument **suffix = c("_emp“,”_mgr")** inside **left_join()** after specifying the **by** argument. Rename **tenure** to **tenure_mgr** and **effectiveness** to **effectiveness_mgr**. Use **mutate_at()** to adjust **career_satisfaction**, **perf_satisfaction**, and **work_satisfaction** by multiplying them by *100*. You will use `~ 100*.` as the second input inside **mutate_at()**. Note, you will use what is between the two backticks (i.e., you will use: tilde, 100, asterisk, dot).

Question 3.1: After joining these three tables, how many variables are in **org_data**?

Response 3.1: *28.*

Recode the following listed variables in **org_data** using **mutate_at()**, **mutate()**, and **fct_recode()** or **factor()** as appropriate:

1. **turnover** - nominal factor: 0 = “Active”, 1 = “Inactive”
2. **location** - nominal factor: 1 = “New York”, 2 = “Chicago”, 3 = “Orlando”
3. **level** - nominal factor: 1 = “Analyst”, 2 = “Specialist”
4. **gender** - nominal factor: 1 = “Female”, 2 = “Male”
5. **hiring_source** - nominal factor: 1 = “Consultant”, 2 = “Job Fairs”, 3 = “Job Boards”, 4 = “Social Media”, 5 = “Walk-In”, 6 = “Employee Referral”, 7 = “Company Website”
6. **marital_status** - nominal factor: 1 = “Single”, 2 = “Married”
7. **education** - nominal factor: 1 = “Bachelors”, 2 = “Masters”
8. **promotion_last_2_years** - nominal factor: 1 = “No”, 2 = “Yes”
9. **rating_emp** - ordered factor: 1 = “Unacceptable”, 2 = “Below Average”, 3 = “Acceptable”, 4 = “Above Average”, 5 = “Excellent”
10. **rating_mgr** - ordered factor: 1 = “Unacceptable”, 2 = “Below Average”, 3 = “Acceptable”, 4 = “Above Average”, 5 = “Excellent”

Note, only **rating_emp** and **rating_mgr** should be treated as *ordered* factor variables.

As a hint: You can use *two* **mutate_at** statements to convert numeric variables to nominal and ordered factor variables, respectively. Then, you can use a **mutate** statement to assign the category labels for the eight nominal factor variables. Then, you can use a **mutate_at** statement to assign the category labels for the two ordered factor variables.

Use **glimpse** on **org_data** after completing the mutations.

Question 3.2: What is the **location**, **gender**, **rating_emp**, and **hiring_source** of the *first* employee? What is the **marital_status**, **education**, **promotion_last_2_years**, and **turnover** of the *second* employee?

Response 3.2: *First: New York, Female, Above Average, Consultant. Second: Single, Bachelors, No, Active.*

```r
#### Q3.1
### join tables
org_data <- emp_mgr_ids %>%
  ## join emp_mgr_ids with employees
  left_join(employees, by = c("emp_id" = "id"))%>%
  ## join managers
  left_join(managers, by = c("mgr_id" = "id"), suffix = c("_emp", "_mgr"))  %>%
  ## rename joined variables
  rename(tenure_mgr = tenure, effectiveness_mgr = effectiveness)

org_data <- org_data %>%
        ## compute new factor variable from existing factor variable
  mutate_at(c("career_satisfaction", "perf_satisfaction", "work_satisfaction"), ~ 100*., na.rm=TRUE)

#### Q3.2
###transform data
org_data <- org_data %>%
  ##select nominal factors
  mutate_at(vars(turnover,location, level, gender, hiring_source, marital_status, education, promotion_l
  ##select ordered factors
  mutate_at(vars(rating_emp, rating_mgr), factor, ordered=TRUE) %>%
  ##recode factor
  mutate_at(vars(rating_emp, rating_mgr),
            ~fct_recode(., `Unacceptable` = "1", `Below Average` = "2", `Acceptable` = "3", `Above Avera

#recode levels for turnover
org_data <- org_data %>%
  mutate(turnover = fct_recode(turnover,
                      # change 0 to Active
                      `Active` = "0",
                      # change 1 to Inactive
                      `Inactive` = "1"))

#recode levels for location factor
org_data <- org_data %>%
  mutate(location = fct_recode(location,
                      # change 1 to New York
                      `New York` = "1",
                      # change 2 to Chicago
                      `Chicago` = "2",
                      #change 3 to Orlando
                      `Orlando` = "3"))

#recode levels for level factor
org_data <- org_data %>%
  mutate(level = fct_recode(level,
                      # change 1 to Analyst
                      `Analyst` = "1",
                      # change 2 to Specialist
                      `Specialist` = "2"))

#recode levels for gender factor
org_data <- org_data %>%
```

```r
  mutate(gender = fct_recode(gender,
                      #change 1 to Female
                      `Female` = "1",
                      #change 2 to Male
                      `Male` = "2"))

#recode levels for hiring_source factor
org_data <- org_data %>%
  mutate(hiring_source = fct_recode(hiring_source,
                      #change 1 to Consultant
                      `Consultant` = "1",
                      #change 2 to Job Fairs
                      `Job Fairs` = "2",
                      #change 3 to Job Boards
                      `Job Boards` = "3",
                      #change 4 to Social Media
                      `Social Media` = "4",
                      #change 5 to Walk-In
                      `Walk-In` = "5",
                      #change 6 to Employee Referral
                      `Employee Referral` = "6",
                      #change 7 to Company Website
                      `Company Website` = "7"))

#recode levels for marital_status factor
org_data <- org_data %>%
  mutate(marital_status = fct_recode(marital_status,
                      #change 1 to Single
                      `Single` = "1",
                      #change 2 to Married
                      `Married` = "2"))

#recode levels for education factor
org_data <- org_data %>%
  mutate(education = fct_recode(education,
                      #change 1 to Bachelors
                      `Bachelors` = "1",
                      #change 2 to Masters
                      `Masters` = "2"))

#recode levels for promotion_last_2_years factor
org_data <- org_data %>%
  mutate(promotion_last_2_years = fct_recode(promotion_last_2_years,
                      #change 1 to No
                      `No` = "1",
                      #change 2 to Yes
                      `Yes` = "2"))

### using glimpse
glimpse(org_data)
```

```
## Rows: 1,954
## Columns: 28
```

```
## $ emp_id                      <chr> "E10012", "E10025", "E10027", "E10048", "~
## $ mgr_id                      <chr> "E9335", "E6655", "E13942", "E7063", "E56~
## $ location                    <fct> New York, Chicago, Orlando, Chicago, Orla~
## $ level                       <fct> Analyst, Analyst, Specialist, Specialist,~
## $ gender                      <fct> Female, Female, Female, Male, Male, Male,~
## $ age_emp                     <dbl> 25.09, 25.98, 33.40, 24.55, 31.23, 31.98,~
## $ rating_emp                  <ord> Above Average, Acceptable, Acceptable, Ac~
## $ compensation                <dbl> 64320, 48204, 85812, 49536, 75576, 56904,~
## $ percent_hike                <dbl> 10, 8, 11, 8, 12, 8, 12, 9, 9, 6, 11, 7, ~
## $ hiring_score                <dbl> 70, 70, 77, 71, 70, 75, 72, 70, 70, 70, 7~
## $ hiring_source               <fct> Consultant, Job Fairs, Consultant, Job Bo~
## $ no_previous_companies_worked <dbl> 0, 9, 3, 5, 0, 8, 9, 6, 1, 3, 3, 6, 2, 6,~
## $ distance_from_home          <dbl> 14, 21, 15, 9, 25, 23, 17, 16, 22, 22, 18~
## $ total_dependents            <dbl> 2, 2, 5, 3, 4, 5, 2, 5, 2, 5, 5, 5, 4, 5,~
## $ marital_status              <fct> Single, Single, Single, Single, Single, S~
## $ education                   <fct> Bachelors, Bachelors, Bachelors, Bachelor~
## $ promotion_last_2_years      <fct> No, No, Yes, Yes, No, No, No, No, No, No,~
## $ no_leaves_taken             <dbl> 2, 10, 18, 19, 25, 15, 10, 20, 22, 23, 24~
## $ total_experience            <dbl> 6.86, 4.88, 8.55, 4.76, 8.06, 13.72, 5.81~
## $ monthly_overtime_hrs        <dbl> 1, 5, 3, 8, 1, 7, 2, 10, 2, 10, 8, 3, 1, ~
## $ turnover                    <fct> Active, Active, Active, Active, Active, I~
## $ career_satisfaction         <dbl> 73, 72, 85, 42, 78, 88, 68, 76, 33, 100, ~
## $ perf_satisfaction           <dbl> 73, 84, 80, 33, 67, 81, 57, 74, 50, 80, 2~
## $ work_satisfaction           <dbl> 75, 85, 87, 85, 80, 86, 75, 95, 87, 88, 7~
## $ rating_mgr                  <ord> Acceptable, Excellent, Above Average, Acc~
## $ age_mgr                     <dbl> 44.07, 35.99, 35.78, 26.70, 34.28, 34.82,~
## $ tenure_mgr                  <dbl> 3.17, 7.92, 4.38, 2.87, 12.95, 10.88, 4.0~
## $ effectiveness_mgr           <dbl> 0.730, 0.581, 0.770, 0.240, 0.710, 0.574,~
```

Task 4: Plot Categorical Variables

Use **ggplot** to produce a *horizontal percentage bar plot* of **hiring_source** such that the most frequent category is the top bar in the plot and the least frequent category is the bottom bar in the plot. You will need to use **y = fct_rev(fct_infreq(hiring_source))** inside the **aes()** statement of **ggplot()**.
The x-axis will represent *percent formatted* values. The y-axis will represent the different categories of **hiring_source**. Label the x and y axes appropriately.

Question 4.1: What is the most frequent source of hiring? What is the least frequent source of hiring?

Response 4.1: *Most frequent: Consultant. Least frequent: Employee Referral.*

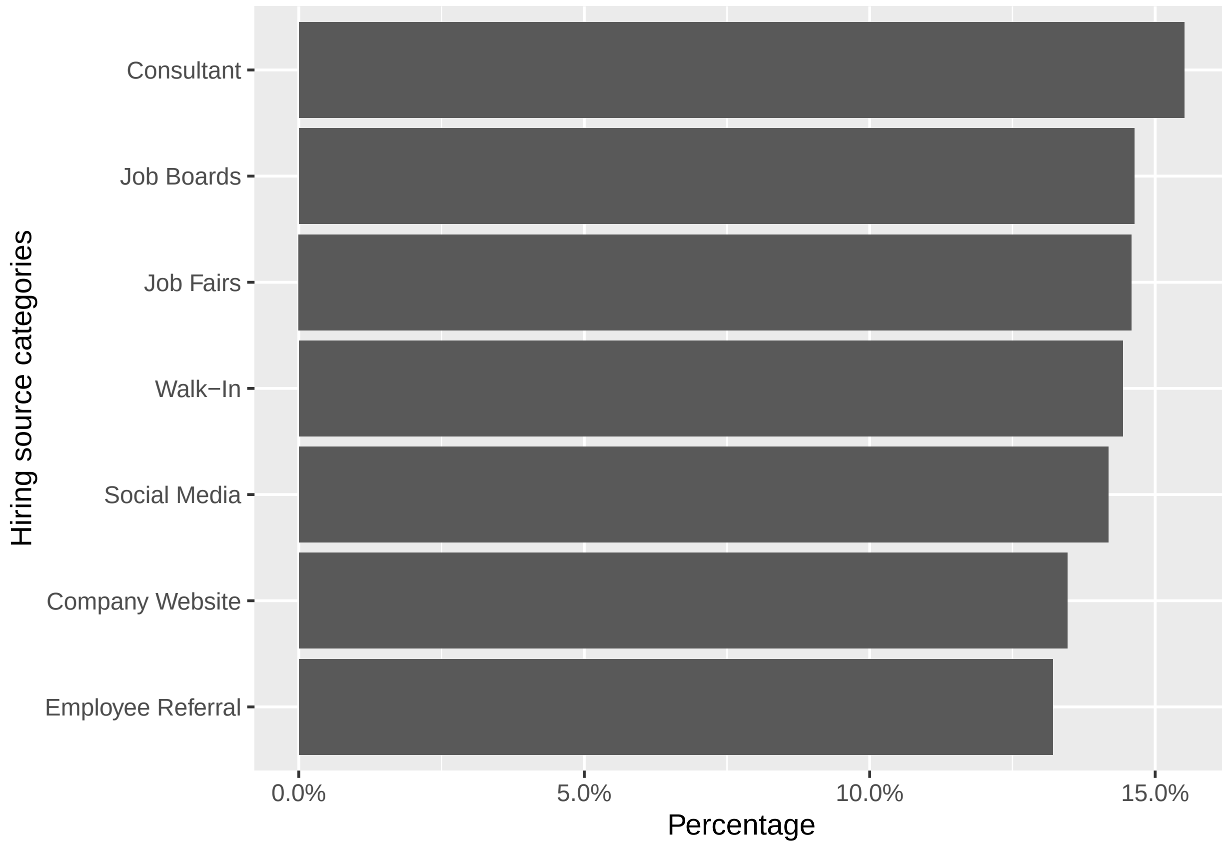
Use **ggplot** to produce a *vertical percentage bar plot* of **hiring_source** by filled by **gender**. The x-axis will represent **hiring_source**. The bars will be colored by **gender**. The y-axis will represent percentage of men and women per hiring source. Label the top of each bar with the actual percentage value. Angle the text on the x-axis at 45 degrees. Title the plot: *Percentage of Men and Women per Hiring Source.*

Question 4.2: Which hiring source is most frequent for *women*? Which hiring source is most frequent for *men*?

Response 4.2: *Women: Social Media. Men: Consultant.*

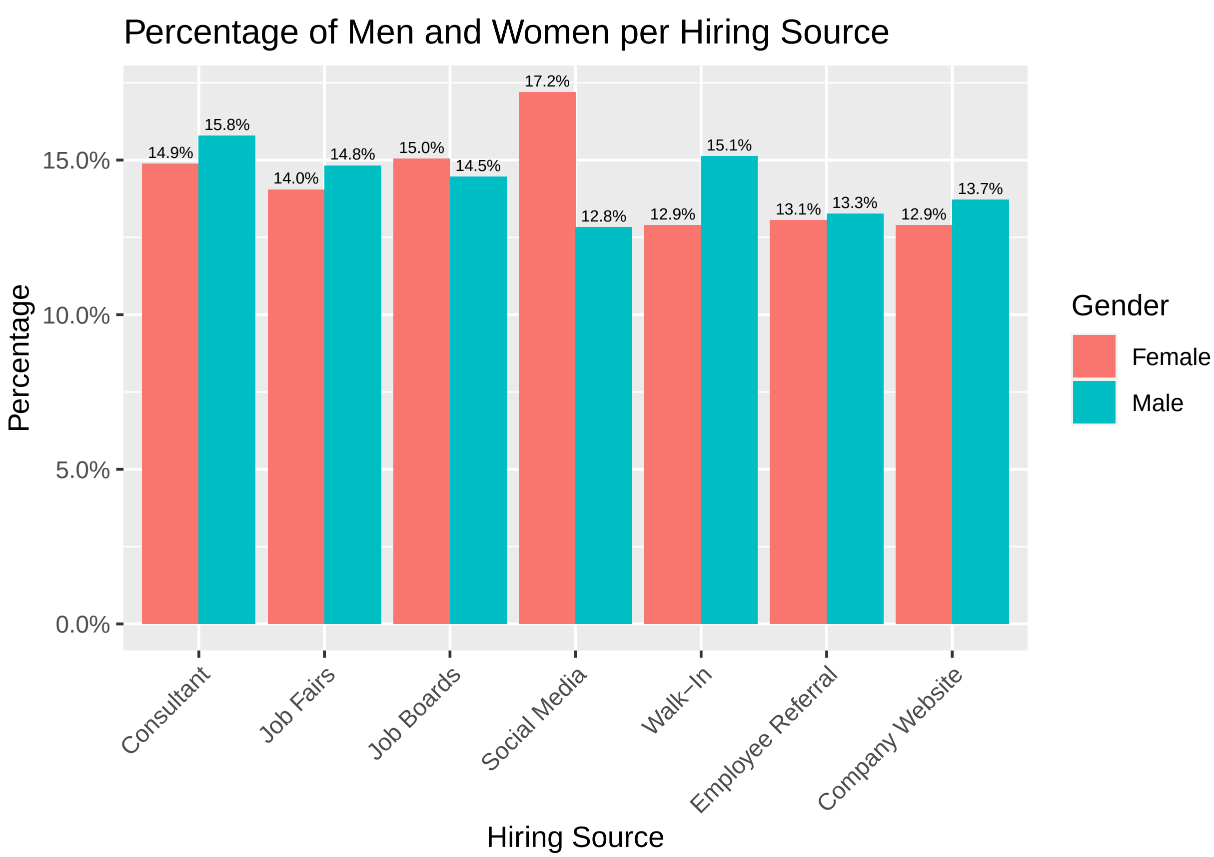
```r
#### Q4.1
### plot single categorical variable
## choose data and mapping
ggplot(data = org_data, mapping = aes(y = fct_rev(fct_infreq(hiring_source)))) +
  ## choose geometry with proportion calculation
```

```
  geom_bar(aes(x = ..prop.., group = 1)) +
  ## label axes
  labs(y = "Boss Gender", x = "Percentage") +
  ## change format of x-axis
  scale_x_continuous(labels = scales::percent_format())+
  xlab("Percentage") + ylab("Hiring source categories")
```




```
#### Q4.2
### plot multiple categorical variables;
## choose data
ggplot(data = org_data %>%
         # filter for only non-missing values
         filter(!is.na(gender)),
       ## specify mapping
       mapping = aes(x = hiring_source,
                     group = gender,
                     fill = gender)) +
  ## choose geometry with proportion calculation
  geom_bar(aes(y = ..prop..),
           position = "dodge") +
  ## label mappings
  labs(x = "Hiring Source", y = "Percentage", fill = "Gender") +
  ## change format of y-axis
  scale_y_continuous(labels = scales::percent_format()) +
  ## add text above bars
```

```r
            # stat for geometry
  geom_text(stat = "count",
                # location of label
            aes(y = ..prop..,
                # label and number of digits
                label = scales::percent(..prop.., accuracy = 0.1)),
            # justify vertically above bar
            vjust = -0.5,
            # position label above each bar
            position = position_dodge(0.9), size = 2) +
  ## change angle of x-axis labels
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ## add descriptive title
  ggtitle("Percentage of Men and Women per Hiring Source")
```



Task 5: Plotting Continuous Variables

Use **ggplot** to show the boxplots for **level** on **career_satisfaction** faceted by **education** in the rows and **gender** in the columns. Color outliers *red*. Label the y-axis: *Career Satisfaction Percentile*. Label the x-axis: *Job*. Remove the legend. Fill each set of boxplots differently in each cell of the facet grid as a function of the grid variables (i.e., **education** and **gender**). You will need to use **interaction** to accomplish this last aesthetic.

Question 5.1: Which combination of **gender**, **level**, and **education** has the lowest median **career_satisfaction** score?

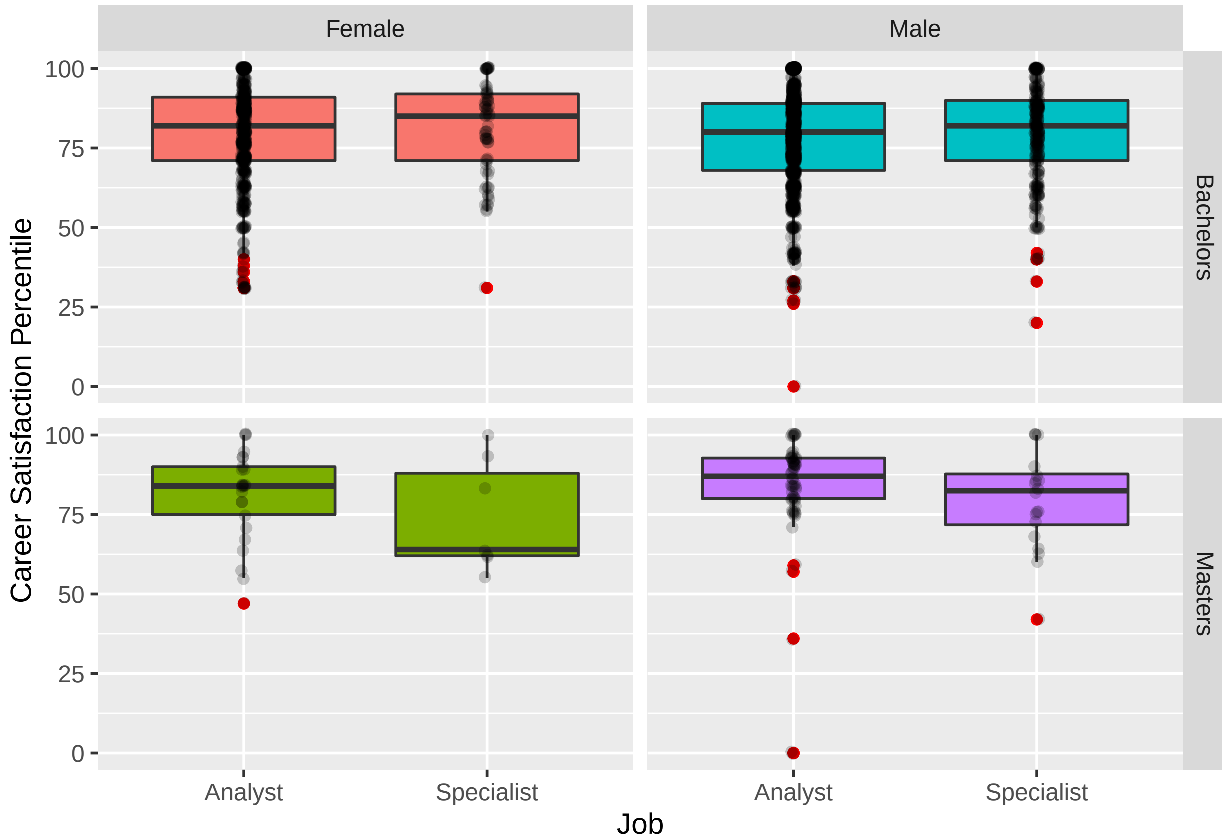
Response 5.1: *Female Specialist with a Masters degree.*

Use **ggplot** to show the scatterplot between **perf_satisfaction** and **career_satisfaction**. Place **perf_satisfaction** on the x-axis and **career_satisfaction** on the y-axis. Color the data points by **gender** Use **geom_jitter** and not **geom_point**. Fit the *loess* line through the data points. Label the x-axis: *Performance Satisfaction*. Label the y-axis: *Career Satisfaction*. Label the legend: *Gender*. Use the *Dark2* color palette via **scale_color_brewer**.

Question 5.2: Is the overall relationship between **perf_satisfaction** and **career_satisfaction** positive or negative? Is the relationship between the two variables quite similar or different when comparing men and women?

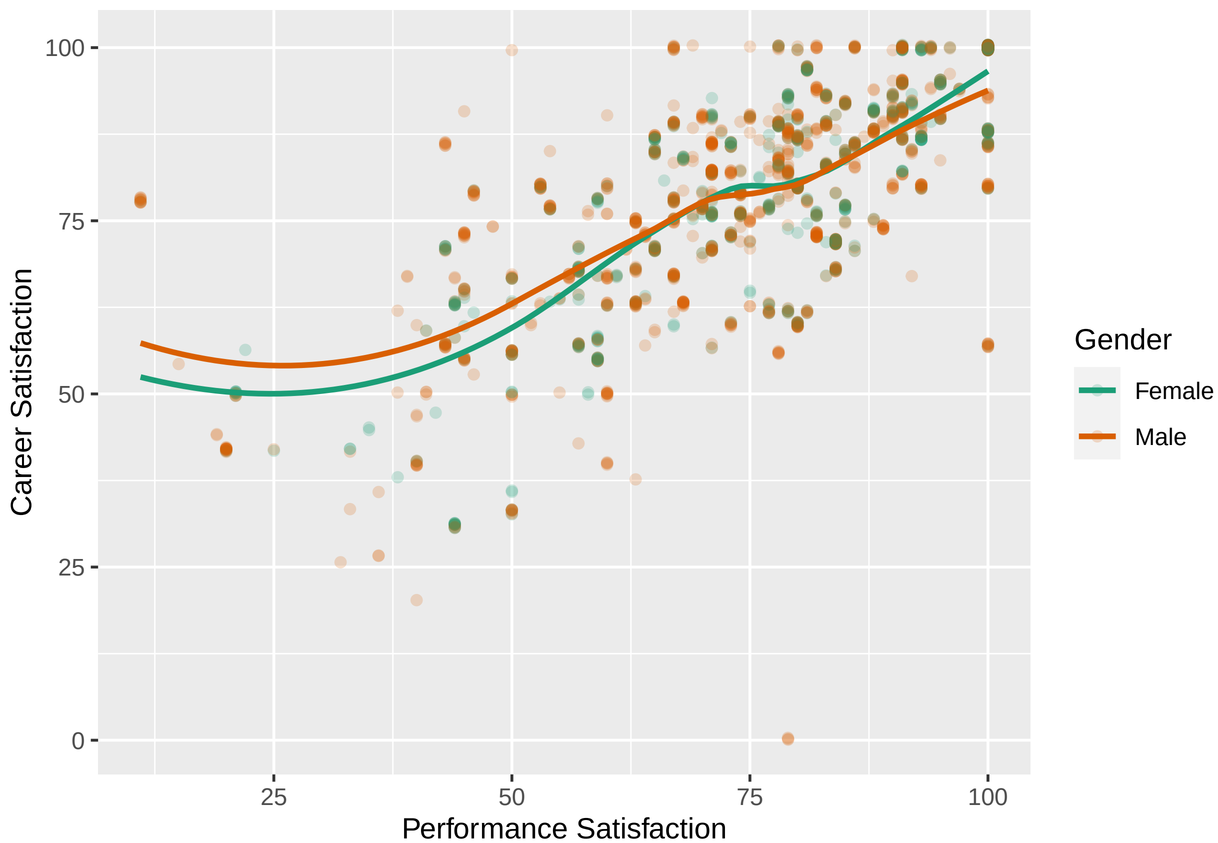
Response 5.2: *The overall relationship between perf_satisfaction and career_satisfaction is positive. The relationship between the two variables is quite similar when comparing men and women.*

```
#### Q5.1
### create boxplot
## choose data and mapping
ggplot(data = org_data,
       # place level and career_satisfaction on x and y axes
       mapping = aes(x = level, y = career_satisfaction,
                     # color boxplots by education and gender
                     # simultaneously
                     fill = interaction(education, gender))) +
  ## add boxplot
  geom_boxplot(outlier.color = "red") +
  ## add points
  geom_jitter(width = 0.01, alpha = 0.2) +
  ## facet for variable type
  facet_grid(education ~ gender) +
  ## labels
  labs(x= "Job", y = "Career Satisfaction Percentile") +
  ## hide legend
  theme(legend.position = "none")
```




```
#### Q5.2
### examine relationship between two numeric variables;
### use loess line to examine type of relationship;
### use factor variable to color points
## choose data
ggplot(org_data, aes(x = perf_satisfaction, y = career_satisfaction,
                            # color data points
                            color = gender)) +
  ## choose point geometry for scatterplot
  geom_jitter(width = 0.01, alpha = 0.2) +
  ## loess line
  geom_smooth(method = "loess", se = FALSE) +
  ## label axes
  labs(x = "Performance Satisfaction", y = "Career Satisfaction", color = "Gender") +
  ## change default colors
  scale_color_brewer(palette = "Dark2")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Task 6: Correlations and Distances

Use **ggpairs** to produce a scatterplot matrix for **compensation**, **hiring_score**, **total_experience**, **career_satisfaction**, **perf_satisfaction**, and **work_satisfaction**. Make sure to use **dplyr::select()** when selecting variables.

Question 6.1: What is the largest correlation in the matrix? What does the small correlation between **career_satisfaction** and **compensation** conceptually indicate?

Response 6.1: *0.695 (correlation between career_satisfaction and perf_satisfaction). The small correlation between career_satisfaction and compensation conceptually indicates that these two variables have a weak linear relationship.*

Compute a new object named **comp_means** where you group **org_data** by **level**, **gender**, and **education** simultaneously in that order. Then, apply **skim_without_charts()** and **filter()** by **skim_variable == "compensation"**. Print **comp_means** to see the results.

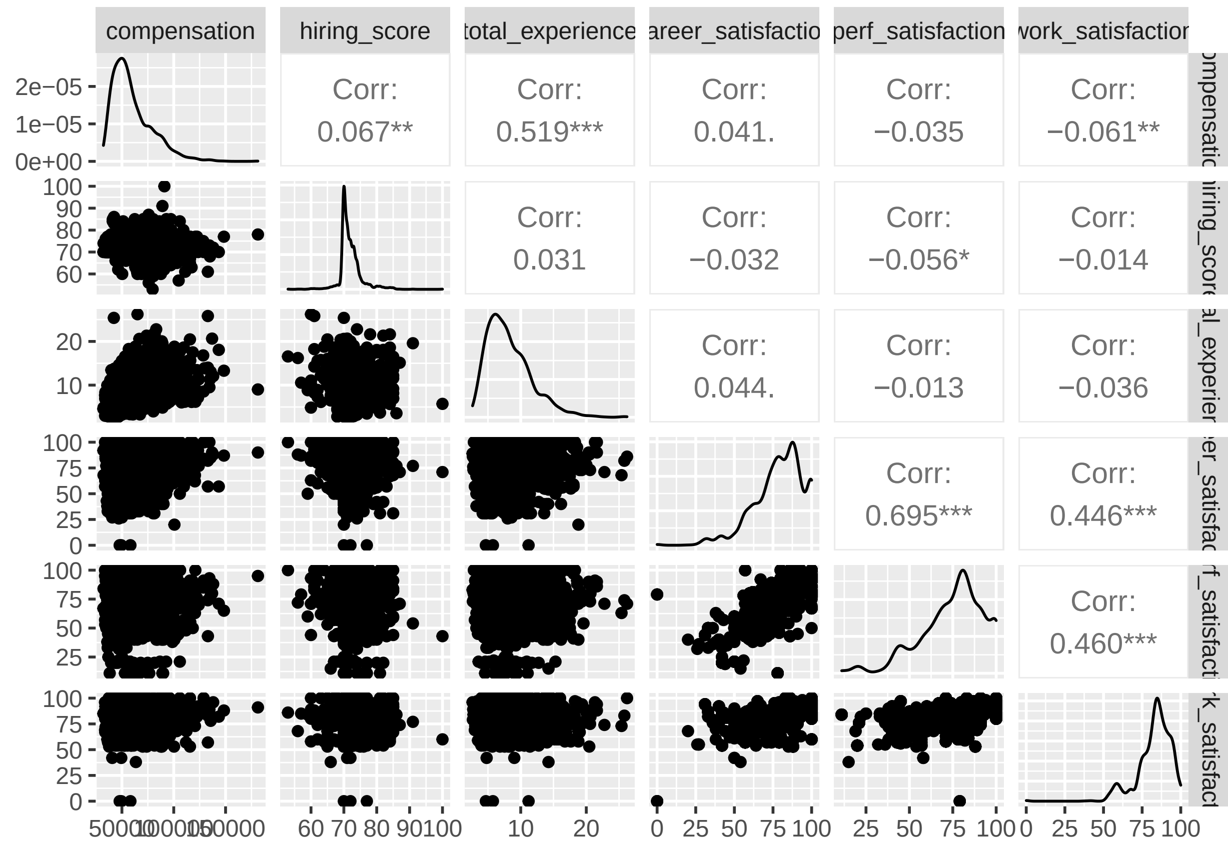
Next, compute a new object named **comp_dist_means** selecting **numeric.mean** (use **dplyr::select()**), computing the *Manhattan* distance, converting the result to a matrix, and applying **sqrt()** to all the values. Name the rows and columns of **dist_means** with the following vector: **c("afb", "afm", "amb", "amm", "sfb", "sfm", "smb", "smm")**. The first letter identifies whether the person is an analyst (a) or specialist (s). The second letter identifies whether the person is female (f) or male (m). The third letter identifies whether the person has a Bachelor's (b) or Master's (m) degree. Print **comp_dist_means** to see the result.

Apply **qgraph()** to **comp_dist_means** with the **spring** layout.

Question 6.2: Which two groups differ the most with respect to **compensation**? Is the Bachelor's educated female analyst (**afb**) more similar on **compensation** with the Master's educated male analyst (**amm**) or the Bachelor's educated female specialist (**sfb**)?

Response 6.2: *Groups afb and sfm differ the most. Afb is more similar to amm than it is to sfb.*

```
#### Q6.1
### scatterplot matrix
## choose data
org_data %>%
  ## select variables
  dplyr::select(compensation, hiring_score, total_experience, career_satisfaction, perf_satisfaction, wo
  ## scatterplot matrix
  ggpairs()
```




```
#### Q6.2
### distances between groups
## choose data
comp_means <- org_data %>%
  ## grouping variables
  group_by(level, gender, education) %>%
  ## summary
  skim_without_charts() %>%
  ## filter
  filter(skim_variable == "compensation")
```

```r
##print comp_means
comp_means
```

Table 1: Data summary

Name	Piped data
Number of rows	1954
Number of columns	28
Column type frequency:	
numeric	1
Group variables	level, gender, education

Variable type: numeric

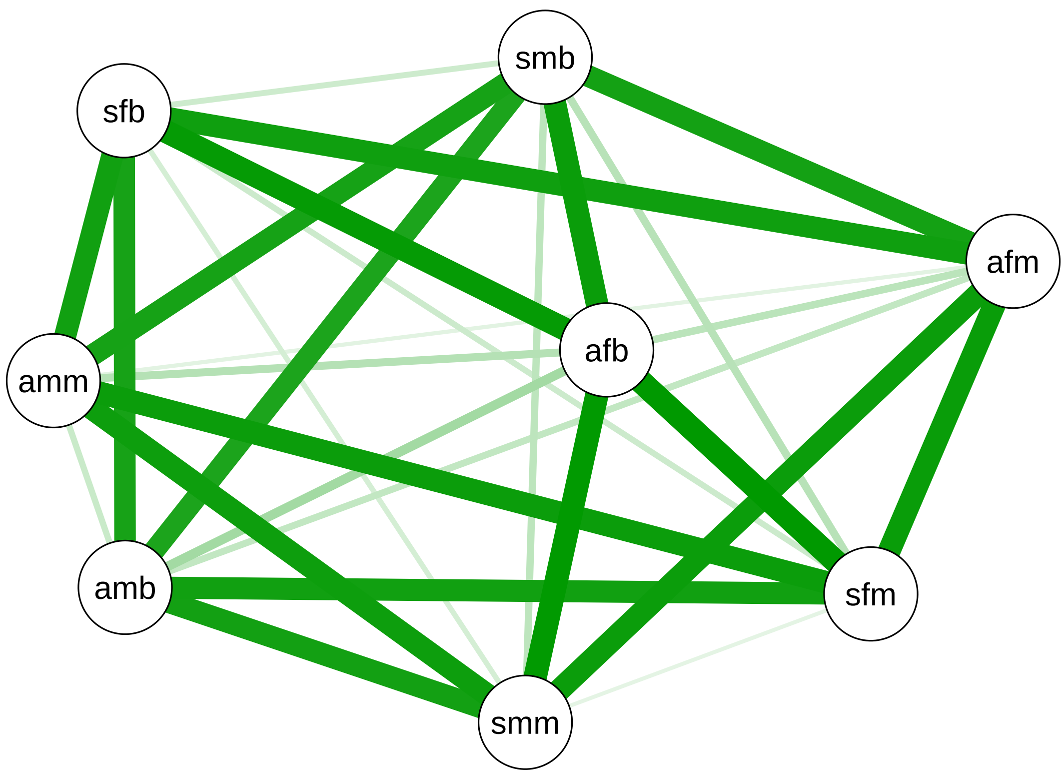
skim_variable	level	gender	education	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
compensation	Analyst	Female	Bachelors	0	1	52018.66	12728.84	33696	43155	50010	57228	120864
compensation	Analyst	Female	Masters	0	1	54396.48	14175.60	33768	43104	51396	63564	84444
compensation	Analyst	Male	Bachelors	0	1	56301.90	14860.86	32304	44916	53004	65142	137004
compensation	Analyst	Male	Masters	0	1	54832.45	16206.89	32148	43101	49650	64503	92784
compensation	Specialist	Female	Bachelors	0	1	83820.51	26990.92	42480	63750	80292	97260	181212
compensation	Specialist	Female	Masters	0	1	85167.43	24959.34	40584	76020	90444	98562	115980
compensation	Specialist	Male	Bachelors	0	1	82565.43	20141.18	40620	65235	83706	94761	148404
compensation	Specialist	Male	Masters	0	1	84784.20	14432.91	59592	71826	86694	97206	104736

```r
## compute distance matrix
comp_dist_means <- comp_means %>%
  ## select means variable
  dplyr::select(numeric.mean) %>%
  ## compute distance
  dist(method = "manhattan") %>%
  ## convert to matrix
  as.matrix() %>% sqrt()

## name columns
colnames(comp_dist_means) <- row.names(comp_dist_means) <- c("afb", "afm", "amb", "amm", "sfb", "sfm",

##print comp_dist_means
comp_dist_means
```

```
##           afb       afm       amb       amm       sfb       sfm       smb
## afb   0.00000  48.76288  65.44642  53.04517 178.33072 182.06803 174.77633
## afm  48.76288   0.00000  43.65107  20.87993 171.53433 175.41650 167.83608
## amb  65.44642  43.65107   0.00000  38.33333 165.88735 169.89860 162.06028
## amm  53.04517  20.87993  38.33333   0.00000 170.25879 174.16939 166.53221
## sfb 178.33072 171.53433 165.88735 170.25879   0.00000  36.70043  35.42709
## sfm 182.06803 175.41650 169.89860 174.16939  36.70043   0.00000  51.00980
## smb 174.77633 167.83608 162.06028 166.53221  35.42709  51.00980   0.00000
```

```
## smm 181.01254 174.32074 168.76701 173.06573  31.04340  19.57622  47.10384
##           smm
## afb 181.01254
## afm 174.32074
## amb 168.76701
## amm 173.06573
## sfb  31.04340
## sfm  19.57622
## smb  47.10384
## smm   0.00000
```

```r
## plot
qgraph(comp_dist_means, layout = "spring")
```



Task 7: OLS Regression

Build an ordinary least-squares (OLS) multiple regression model where you predict **work_satisfaction** from **perf_satisfaction**, **career_satisfaction**, **gender**, **education**, and **promotion_last_2_years**. Name the model object **mod_1**. Apply **summary()** to **mod_1**. Apply **calc.relimp()** to **mod_1**.

Question 7.1: According the model results, do men or women experience more **work_satisfaction**? How do you interpret the regression coefficient for **career_satisfaction**? Which two variables in the model are most important to predicting **work_satisfaction**?

Response 7.1: *Women experience more work_satisfaction. The regression coefficient for career_satisfaction shows the expected increase in work_satisfaction for one unit increase in ca-*

reer_satisfaction, holding other predictors constant. For one unit increase in career_satisfaction, with the remaining predictors remaining constant, we expect a 0.17119 increase in work_satisfaction. The two most important variables to predicting work_satisfaction are perf_satisfaction and career_satisfaction.

Apply **augment()** to **mod_1** and save the resulting object as **mod_1_fit**. Consider the use of **mod_1** to make a decision on whose job should be redesigned as a function of predicted **work_satisfaction**. The goal is to evaluate how successful **mod_1** is in predicting *low* **work_satisfaction**. Set a predicted work satisfaction threshold variable named **pred_thresh** to 75. Set a real work satisfaction threshold variable named **crit_thresh** to 80.

Use **ggplot** to show the scatterplot between **.fitted** and actual **work_satisfaction** values using **mod_1_fit**. Show the **pred_thresh** value as a *green* vertical line in the plot. Show the **crit_thresh** value as a *red* horizontal line in the plot.

Calculate the number of true positive, true negative, false positive, and false negative decisions. Save the result as **mod_1_acc**. Print **mod_1_acc**. Then, calculate the positive, negative, sensitivity, and specificity accuracy. In this case, we are most interested in true and false negatives, and, therefore, negative accuracy.

Question 7.2: How many true and false negative decisions would be made using **mod_1** and these thresholds? What is the negative accuracy? Should we use this model and these thresholds to redesign jobs for those with *low* **work_satisfaction** (i.e., is the negative accuracy far greater than 50% or not)?

Response 7.2: *True negative: 89. False negative: 76. Negative accuracy: 0.539. Since the negative accuracy is not far greater than 50%, this model and these thresholds should not be used to redesign jobs for those with low work_satisfaction.*

```r
#### Q7.1
### multiple OLS regression model
## contrasts for gender
contrasts(org_data$gender)
```

```
##        Male
## Female    0
## Male      1
```

```r
## build model
mod_1 <- lm(work_satisfaction ~ perf_satisfaction + career_satisfaction + gender + education + promotio
## summary of results
# fuller output
summary(mod_1)
```

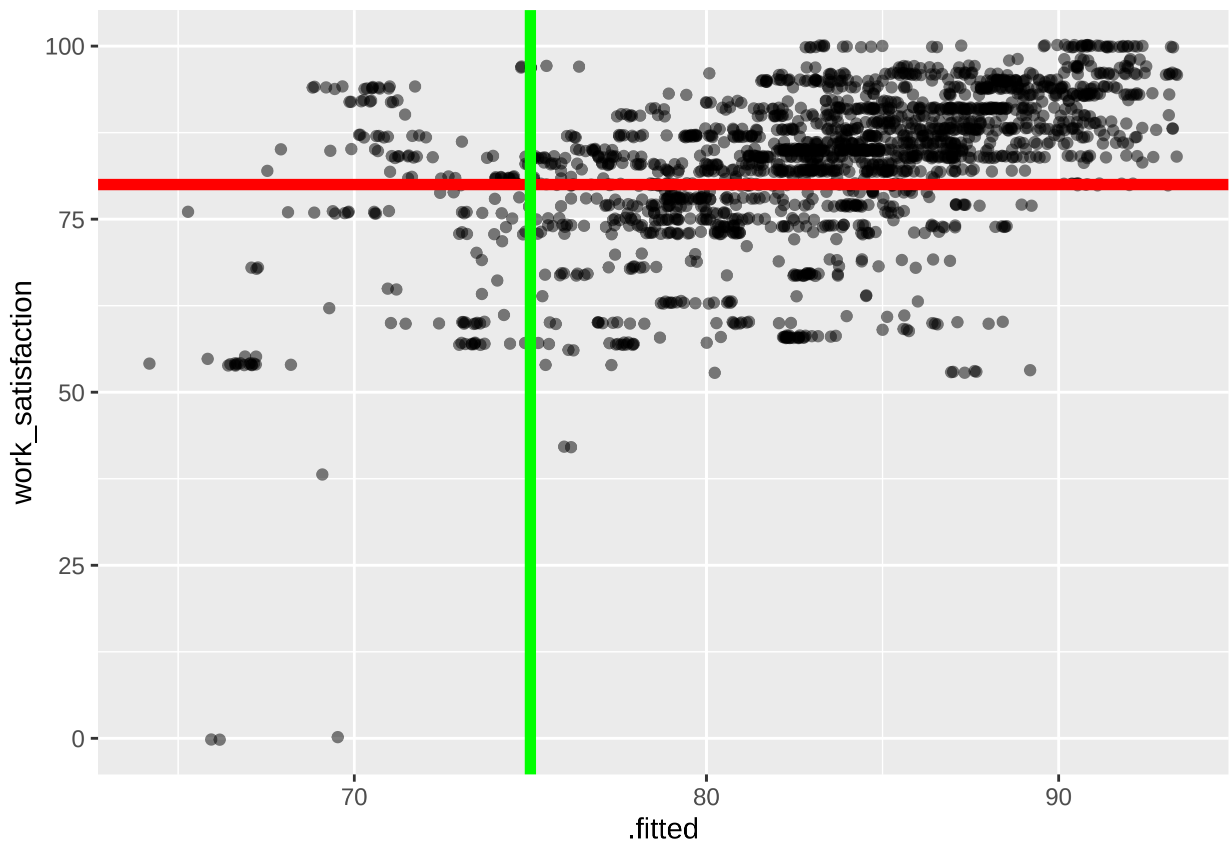
```
##
## Call:
## lm(formula = work_satisfaction ~ perf_satisfaction + career_satisfaction +
##     gender + education + promotion_last_2_years, data = org_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.805  -3.784   1.237   5.863  24.923
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  57.74127    1.16503  49.562  < 2e-16 ***
## perf_satisfaction             0.17243    0.01645  10.481  < 2e-16 ***
```

```
## career_satisfaction        0.17119    0.01888   9.065  < 2e-16 ***
## genderMale                -1.55835    0.46005  -3.387 0.000720 ***
## educationMasters          -3.48089    0.90691  -3.838 0.000128 ***
## promotion_last_2_yearsYes  0.86554    0.50630   1.710 0.087513 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.386 on 1948 degrees of freedom
## Multiple R-squared:  0.2539, Adjusted R-squared:  0.252
## F-statistic: 132.6 on 5 and 1948 DF,  p-value: < 2.2e-16
```

```r
### predictor importance
## specify model and type of relative importance
calc.relimp(mod_1, type = "car")
```

```
## Response variable: work_satisfaction
## Total response variance: 117.7856
## Analysis based on 1954 observations
##
## 5 Regressors:
## perf_satisfaction career_satisfaction gender education promotion_last_2_years
## Proportion of variance explained by model: 25.39%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##                                car
## perf_satisfaction      0.128578969
## career_satisfaction    0.112937488
## gender                 0.005933479
## education              0.004461134
## promotion_last_2_years 0.001998931
```

```r
#### Q7.2
## compute fitted values for all individuals in the sample
mod_1_fit <- augment(mod_1)

### plot data and prediction line
## thresholds
# prediction
pred_thresh <- 75
# criterion
crit_thresh <- 80

## call data and set mapping
ggplot(mod_1_fit, aes(x = .fitted, y = work_satisfaction)) +
  ## jitter geometry
  geom_jitter(width = 0.4, height = 0.2, alpha = 0.5) +
  ## criterion value threshold
  geom_hline(yintercept = crit_thresh, color = "red", size = 2) +
  ## predicted value threshold
  geom_vline(xintercept = pred_thresh, color = "green", size = 2)
```




```
### evaluate accuracy of predictions
## name result and choose data
mod_1_acc <- mod_1_fit %>%
  ## summarize
            # true positives
  summarize(tp = sum(.fitted >= pred_thresh & work_satisfaction >= crit_thresh),
            # true negatives
            tn = sum(.fitted < pred_thresh & work_satisfaction < crit_thresh),
            # false positives
            fp = sum(.fitted >= pred_thresh & work_satisfaction < crit_thresh),
            # false negatives
            fn = sum(.fitted < pred_thresh & work_satisfaction >= crit_thresh))

##print results
mod_1_acc
```

```
## # A tibble: 1 x 4
##      tp    tn    fp    fn
##   <int> <int> <int> <int>
## 1  1346    89   443    76
```

```
## accuracy computations
mod_1_acc %>%
  # overall accuracy
  summarize(overall = (tp + tn)/(tp + tn + fp + fn),
```

```r
            # positive accuracy
            positive = tp/(tp + fp),
            # negative accuracy
            negative = tn/(tn + fn),
            # sensitivity
            sensitivity = tp/(tp + fn),
            # specificity
            specificity = tn/(tn + fp))
```

```
## # A tibble: 1 x 5
##   overall positive negative sensitivity specificity
##     <dbl>    <dbl>    <dbl>       <dbl>       <dbl>
## 1   0.734    0.752    0.539       0.947       0.167
```