Analyze Causality - Causal influence of Self-rated enjoyment of MDS program on expected salary upon graduation

Taiwo Owoseni

Contents

Problem Statement

Using a survey data form 2019 to 2020 of past MDS students

1. Analyse whether a person's self-rated enjoyment of the MDS program (X) had any causal influence on their expected salary upon graduation (Y)?
2. Is there a statistical difference between the data of both years?
3. Does the model without confounding variables perform better than a model with confounding variables

The following are the features in the data set set and their description:

Features	Description
Salary before MDS	Previous salary could be confounding as it will affect future salary expectations and affect happiness in the program. For example, a low prior salary may make a person happier to enter MDS and have a potentially higher future salary.
Years of professional work experience	Years of work experience could be confounding as it will affect future salary expectations. A person with more years of work experience might not be as happy to enter MDS.
Confidence in data science skills before starting MDS	Confidence in data science skills before starting MDS would likely affect future salary expectations and directly influence how happy they were in the program. If a person is more confident, then they will likely have an easier time and be happier.
Frequency of completing optional questions in labs	Doing optional questions in labs indicates a person is likely finishing all of the required content with relative ease or likes a challenge. If this is the case, they could be happier with MDS and expect a larger salary to reflect their skill set.
Actively applying for jobs in block 6 of MDS	If people are actively applying for jobs in block 6 of MDS, it could be affirming that MDS has prepared them adequately or the opposite. This would affect their feelings about the program and, consequently, salary expectations.
Current happiness level	A baseline for happiness. If they are not happy, they are not happy with MDS, and then maybe they do not expect much of a salary.
Frequency of attending MDS career events	Attending career events could increase Confidence in job opportunities and, subsequently, salary. If the events are useful, then the feelings about MDS would likely change for the better.

Note: The columns names in the data sets from *`import_id_qid172807697 to import_id_qid93`are renamed as and* represent the columns *`salary_exp_post_grad`* **(Salary before MDS)** *to* *`freq_attend_mds_career_events`* **(Frequency of attending MDS career events)**

Note: Given that the problem statement can only be realistically answered with an observational study, confounders were considered when collecting the data (via the planned survey) and carrying out the analysis

Exploring Data sets and cleaning

STEP 1: Read data set as csv from local folder, skipped the first two rows.

STEP 2: Clean column names so they are unique and consist only of the _ character, numbers, and letters.

STEP 3: Print out first 5 rows in `salary_df_2019`

```
salary_df_2019 <- read_csv("data/salary_survey_responses_2019.csv", skip = 2) %>%
  clean_names()

salary_df_2020 <- read_csv("data/salary_survey_responses_2020.csv", skip = 2) %>%
  clean_names()

head(salary_df_2019)
```

```
## # A tibble: 6 x 23
##   import_id_start_date_ti~ import_id_end_date~ import_id_status import_id_progr~
##   <dttm>                   <dttm>              <chr>                       <dbl>
## 1 2019-04-03 14:35:00      2019-04-03 14:37:00 IP Address                    100
## 2 2019-04-03 14:38:00      2019-04-03 14:40:00 IP Address                    100
## 3 2019-04-03 14:37:00      2019-04-03 14:39:00 Spam                          100
## 4 2019-04-03 14:53:00      2019-04-03 14:53:00 Survey Preview                100
## 5 2019-04-03 14:54:00      2019-04-03 14:55:00 Spam                          100
## 6 2019-04-03 15:52:00      2019-04-03 15:54:00 IP Address                    100
## # ... with 19 more variables: import_id_duration <dbl>,
## #   import_id_finished <lgl>,
## #   import_id_recorded_date_time_zone_america_denver <dttm>,
## #   import_id_record_id <chr>, import_id_recipient_last_name <lgl>,
## #   import_id_recipient_first_name <lgl>, import_id_recipient_email <lgl>,
## #   import_id_external_data_reference <lgl>,
## #   import_id_distribution_channel <chr>, import_id_user_language <chr>, ...
```

The data was collected **in 2019 and 2020** via a survey with 55 and 44 respondents each.

```
paste('salary_df_2019 has' , nrow(salary_df_2019), 'rows')
```

```
## [1] "salary_df_2019 has 65 rows"
```

```
paste('salary_df_2020 has' , nrow(salary_df_2020), 'rows')
```

```
## [1] "salary_df_2020 has 49 rows"
```

Data Wrangling and Merging data sets **STEP 1**: Renamed data set using the description given in Table 1 as follows

- `import_id_qid172807697` as `salary_exp_post_grad`
- `import_id_qid172807701_1` as `mds_self_rated_enjoy`
- `import_id_qid96` as `salary_pre_mds`
- `import_id_qid172807686` as `work_exp`
- `import_id_qid98_1` as `ds_skill_confidence`
- `import_id_qid172807685` as `does_optional_qs`
- `import_id_qid92` as `currently_job_searching`
- `import_id_qid99_1` as `baseline_happiness`

- `import_id_qid93` as `freq_attend_mds_career_events`

STEP 2: Create a column called `year` with the corresponding labels: `2019` or `2020`.

STEP 3: Drop all observations with missing data.

STEP 4: Merge `salary_df_2019` and `salary_df_2020` into a single data frame called `salary_df`.

STEP 5: Print out first 5 rows in `salary_df`

```
# BEGIN SOLUTION

salary_df_2019 <- salary_df_2019 %>%
  dplyr::select(contains("qid")) %>%
  rename(
    salary_exp_post_grad = import_id_qid172807697,
    mds_self_rated_enjoy = import_id_qid172807701_1,
    salary_pre_mds = import_id_qid96,
    work_exp = import_id_qid172807686,
    ds_skill_confidence = import_id_qid98_1,
    does_optional_qs = import_id_qid172807685,
    currently_job_searching = import_id_qid92,
    baseline_happiness = import_id_qid99_1,
    freq_attend_mds_career_events = import_id_qid93
  ) %>%
  drop_na() %>%
  mutate(year = 2019)

salary_df_2020 <- salary_df_2020 %>%
  dplyr::select(contains("qid")) %>%
  drop_na() %>%
  mutate(year = 2020)
colnames(salary_df_2020) <- colnames(salary_df_2019)

salary_df <- bind_rows(salary_df_2019, salary_df_2020)

head(salary_df)
```

```
## # A tibble: 6 x 10
##   salary_exp_post_grad mds_self_rated_~ salary_pre_mds work_exp ds_skill_confid~
##   <chr>                           <dbl> <chr>          <chr>               <dbl>
## 1 $60,000 to $80,000                  3 $60,000 to $8~ Less th~                2
## 2 $80,001 to $100,000                 3 $60,000 to $8~ 1 - 4 Y~                3
## 3 $80,001 to $100,000                 3 $60,000 to $8~ 4 - 7 Y~                2
## 4 Less than $60,000                   4 Less than $60~ Less th~                4
## 5 $60,000 to $80,000                  3 $60,000 to $8~ 1 - 4 Y~                1
## 6 $80,001 to $100,000                 3 Less than $60~ 1 - 4 Y~                1
## # ... with 5 more variables: does_optional_qs <chr>,
## #   currently_job_searching <chr>, baseline_happiness <dbl>,
## #   freq_attend_mds_career_events <chr>, year <dbl>
```

STEP 1: Get all the unique values for all columns in `salary_df`

```
unique_values<- function(df){
  sapply(df, unique)
}
unique_values(salary_df)
```

```
## $salary_exp_post_grad
## [1] "$60,000 to $80,000"   "$80,001 to $100,000"  "Less than $60,000"
## [4] "More than $120,000"   "$100,001 to $120,000"
##
## $mds_self_rated_enjoy
## [1] 3 4 2 1
##
## $salary_pre_mds
## [1] "$60,000 to $80,000"   "Less than $60,000"    "$100,001 to $120,000"
## [4] "$80,001 to $100,000"  "More than $120,000"
##
## $work_exp
## [1] "Less than 1 Year" "1 - 4 Years"      "4 - 7 Years"      "10+ Years"
## [5] "7 - 10 Years"     "0 - 1 Years"
##
## $ds_skill_confidence
## [1] 2 3 4 1
##
## $does_optional_qs
## [1] "No"  "Yes"
##
## $currently_job_searching
## [1] "Yes" "No"
##
## $baseline_happiness
## [1] 4 2 3 1
##
## $freq_attend_mds_career_events
## [1] "Sometimes" "Not Often" "Often"     "Not often"
##
## $year
## [1] 2019 2020
```

Column Renaming and Value Factoring **STEP 1:** Use `toTitleCase()` to change the level names of `freq_attend_mds_career_events` to *Title Style*.

STEP 2: Change the factor level `0 - 1 Years` to `Less than 1 Year` in `work_exp`.

STEP 3: Convert columns `does_optional_qs`, `currently_job_searching`, and `year` to **NOMINAL factor-type**.

STEP 4: Convert the rest of the columns to **ORDERED factor-type**.

STEP 5: Make sure that factors `salary_exp_post_grad`, `mds_self_rated_enjoy`, `salary_pre_mds`, `work_exp`, `ds_skill_confidence`, `baseline_happiness`, and `freq_attend_mds_career_events` have the correct level order from left to right via function `levels()`. If not, reorder these levels according to the order detailed at the beginning of this exercise.

```
# BEGIN SOLUTION

salary_df <- salary_df %>%
  mutate(
    freq_attend_mds_career_events =
      toTitleCase(freq_attend_mds_career_events)
  ) %>%
  mutate(work_exp = case_when(
```

```
    work_exp == "0 - 1 Years" ~ "Less than 1 Year",
    TRUE ~ work_exp
  )) %>%
  mutate(
    does_optional_qs = factor(does_optional_qs),
    currently_job_searching = factor(currently_job_searching),
    year = factor(year),
    salary_exp_post_grad = factor(salary_exp_post_grad, ordered = TRUE),
    mds_self_rated_enjoy = factor(mds_self_rated_enjoy, ordered = TRUE),
    salary_pre_mds = factor(salary_pre_mds, ordered = TRUE),
    work_exp = factor(work_exp, ordered = TRUE),
    ds_skill_confidence = factor(ds_skill_confidence, ordered = TRUE),
    baseline_happiness = factor(baseline_happiness, ordered = TRUE),
    freq_attend_mds_career_events = factor(freq_attend_mds_career_events,
      ordered = TRUE
    )
  ) %>%
  mutate(
    salary_exp_post_grad = fct_relevel(
      salary_exp_post_grad,
      "Less than $60,000",
      "$60,000 to $80,000",
      "$80,001 to $100,000",
      "$100,001 to $120,000",
      "More than $120,000"
    ),
    salary_pre_mds = fct_relevel(
      salary_pre_mds,
      "Less than $60,000",
      "$60,000 to $80,000",
      "$80,001 to $100,000",
      "$100,001 to $120,000",
      "More than $120,000"
    ),
    work_exp = fct_relevel(
      work_exp,
      "Less than 1 Year",
      "1 - 4 Years",
      "4 - 7 Years",
      "7 - 10 Years",
      "10+ Years"
    ),
    freq_attend_mds_career_events = fct_relevel(
      freq_attend_mds_career_events,
      "Not Often",
      "Sometimes",
      "Often"
    )
  )

# END SOLUTION
```

Previous salary prior to MDS (`salary_pre_mds`) has the following 5 levels:

- Less than $60,000
- $60,000 to $80,000
- $80,001 to $100,000
- $100,001 to $120,000
- More than $120,000

Years of professional work experience prior to MDS (`work_exp`) has the following 5 levels:

- 0 - 1 Years
- 1 - 4 Years
- 4 - 7 Years
- 7 - 10 Years
- 10+ Years

Confidence in data science skill when first starting MDS on a scale of 1 - 4. With 4 being very confident and 1 being not confident. (`ds_skill_confidence`) has the following 4 levels:

- 1
- 2
- 3
- 4

Does optional questions in labs (`does_optional_qs`) has the following 2 levels:

- Yes
- No

Currently applying for data science jobs (`currently_job_searching`) has the following 2 levels:

- Yes
- No

Current happiness level on a scale of 1-4. With 4 being very happy and 1 being not happy. (`baseline_happiness`) has the following 4 levels:

- 1
- 2
- 3
- 4

Frequently attending MDS career events (`freq_attend_mds_career_events`) has the following 3 levels:

- Not often
- Sometimes
- Often

Visualization

Make eight suitable plots of `salary_exp_post_grad` versus each one of the rest of factor-type variables except `year`. Nonetheless, in these eight plots, include panels per `year`.

> **Hint:** If you are using the same class of plot eight times, it would be practical to build a function first.

```
# BEGIN SOLUTION

# Plotting function

plotting_function <- function(dataset, response, exp_variable, x_axis, plot_title) {
  dataset.prop.summary <- dataset %>%
```

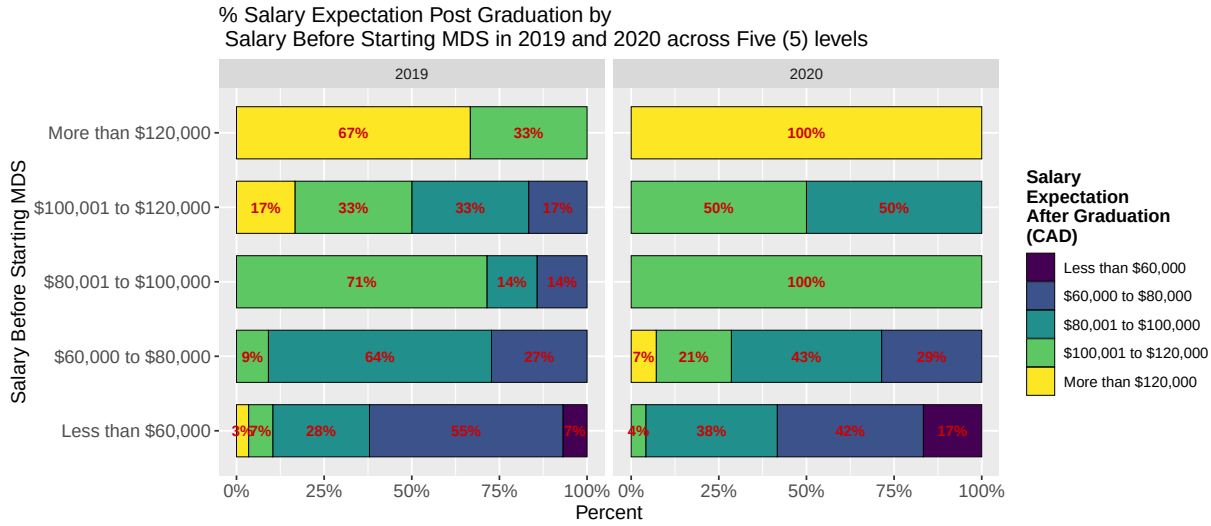
```
    group_by(year, eval(parse(text = exp_variable)), eval(parse(text = response))) %>%
    summarize(n = length(eval(parse(text = exp_variable)))) %>%
    mutate(prop = n / sum(n))

  colnames(dataset.prop.summary)[2:3] <- c(exp_variable, response)

  ggplot(dataset.prop.summary, aes(
    x = eval(parse(text = exp_variable)),
    y = prop, fill = eval(parse(text = response))
  )) +
    ggtitle(plot_title) +
    geom_bar(stat = "identity", width = 0.7, colour = "black", lwd = 0.1) +
    geom_text(aes(label = ifelse(prop >= 0.01,
      paste0(sprintf("%.0f", prop * 100), "%"), ""
    )),
    position = position_stack(vjust = 0.5), colour = "red3",
    fontface = "bold", size = 3
    ) +
    scale_y_continuous(labels = percent_format()) +
    labs(y = "Percent", x = x_axis, fill = "") +
    theme(
      axis.text.x = element_text(size = 11, angle = 0),
      axis.text.y = element_text(size = 11, angle = 0),
      axis.title = element_text(size = 12),
      legend.text = element_text(size = 9, margin = margin(r = 0.5, unit = "cm")),
      legend.title = element_text(size = 11, face = "bold")
    ) +
    guides(fill = guide_legend(title = "Salary\nExpectation\nAfter Graduation\n(CAD)")) +
    facet_grid(~year) +
    coord_flip()
}
```

Percentage of `salary_exp_post_grad` vs `mds_self_rated_enjoy` Regarding the explanatory variable of interest `mds_self_rated_enjoy`, there is a more diverse composition of salary expectation as the level of happiness increases, it leans more towards higher salary expectations.

```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "mds_self_rated_enjoy",
  "Self-Rated Enjoyment", "% Salary Expectation Post Graduation \n by Self Rated Enjoyment in 2019 and 1
)
```

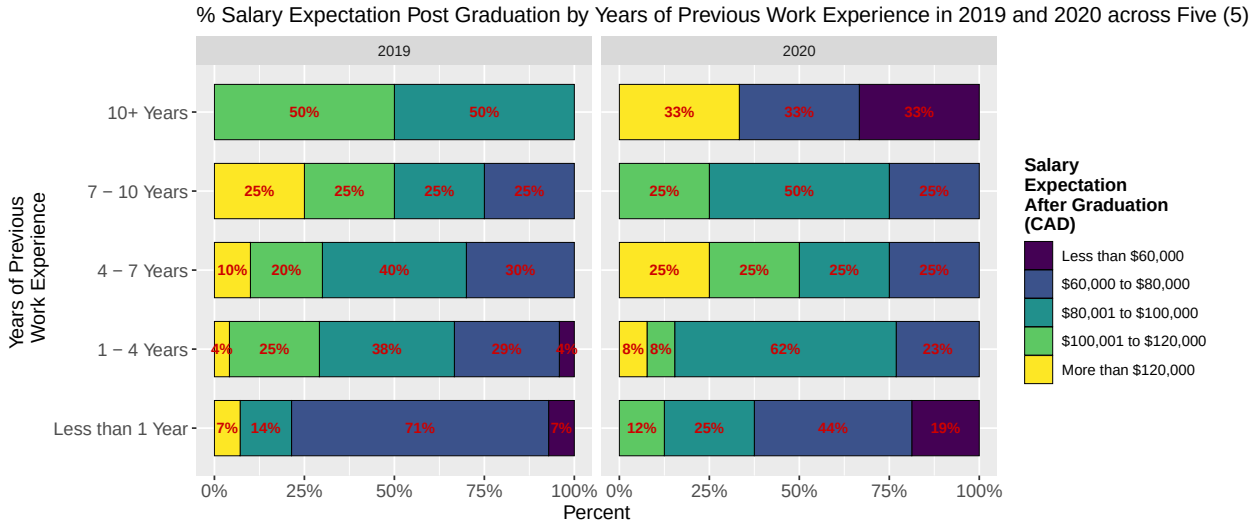
Percentage of `salary_exp_post_grad` vs `salary_pre_mds` In general, regardless of the year, the respondents tend to expect a higher salary once they graduate from MDS compared to their previous salary.

```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "salary_pre_mds",
  "Salary Before Starting MDS", "% Salary Expectation Post Graduation by\n Salary Before Starting MDS i
)
```



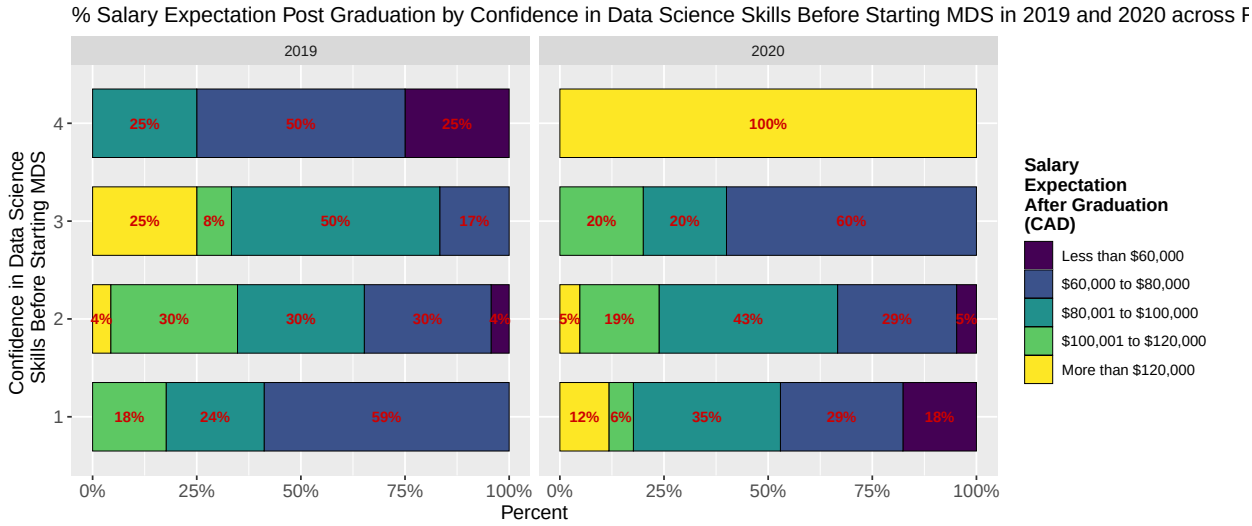
Percentage of `salary_exp_post_grad` vs `work_exp` Graphically, the respondents with more years of experience tend to expect higher salaries once they graduate from MDS. This is more noticeable in 2019 than in 2020.

```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "work_exp",
  "Years of Previous \n Work Experience",  "% Salary Expectation Post Graduation by Years of Previous W
)
```



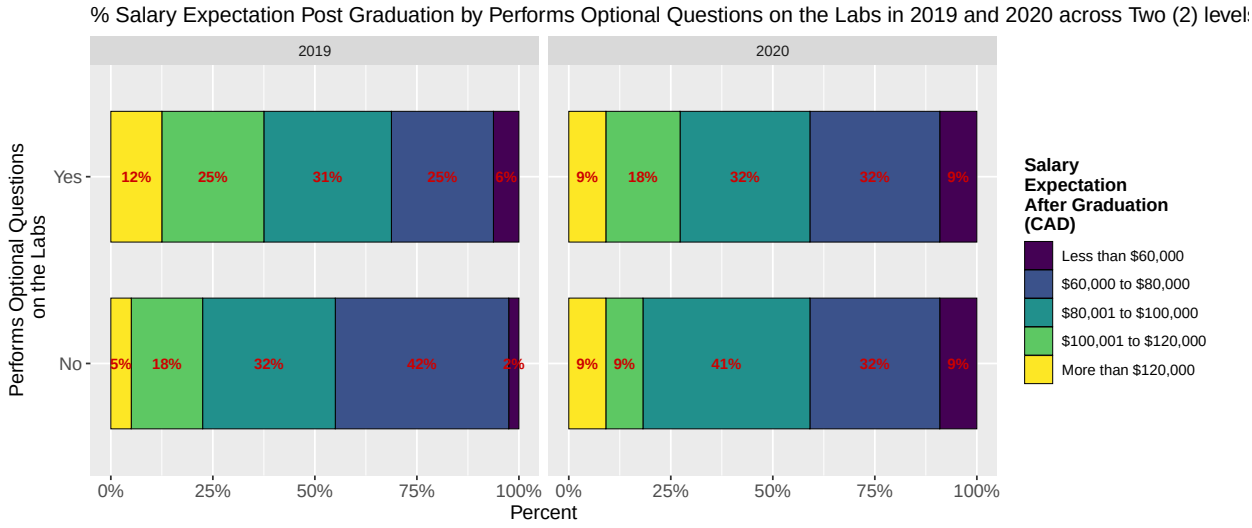
Percentage of `salary_exp_post_grad` vs `ds_skill_confidence` An higher level of confidence before starting MDS is graphically more associated with a higher salary expectation.

```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "ds_skill_confidence",
  "Confidence in Data Science \n Skills Before Starting MDS", "% Salary Expectation Post Graduation by
)
```



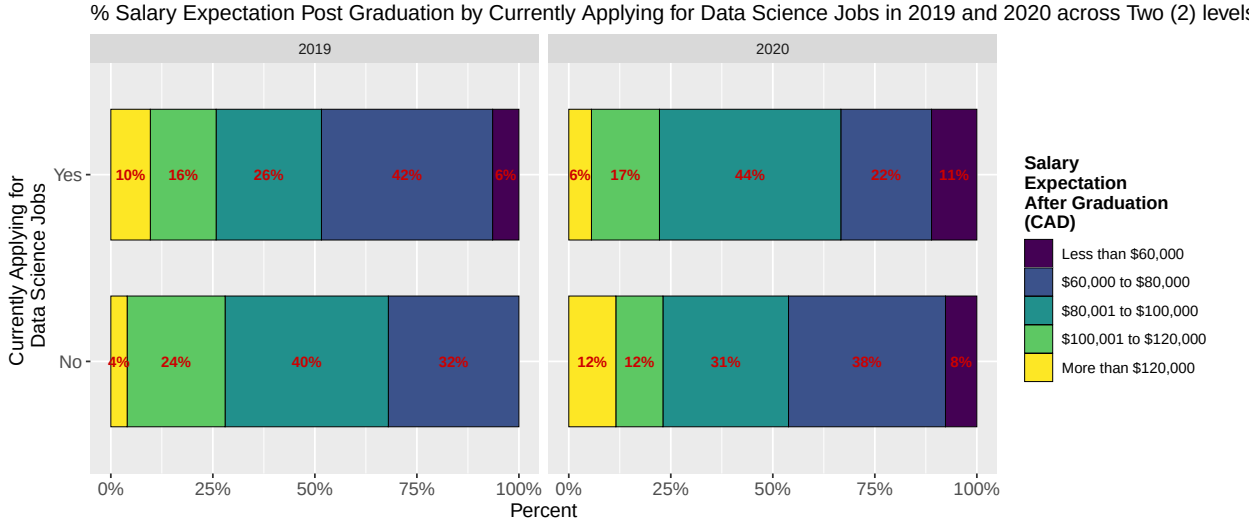
Percentage of `salary_exp_post_grad` vs `does_optional_qs` There are no large graphical differences in the respondent shares between both years. Also, those who perform optional questions also tend to expect a higher salary

```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "does_optional_qs",
  "Performs Optional Questions\n on the Labs",  "% Salary Expectation Post Graduation by Performs Optio
)
```



Percentage of `salary_exp_post_grad` vs `currently_job_searching` The plot does not provide a clear trend between the respondent shares in both levels of the confounder. This also applies between both years.

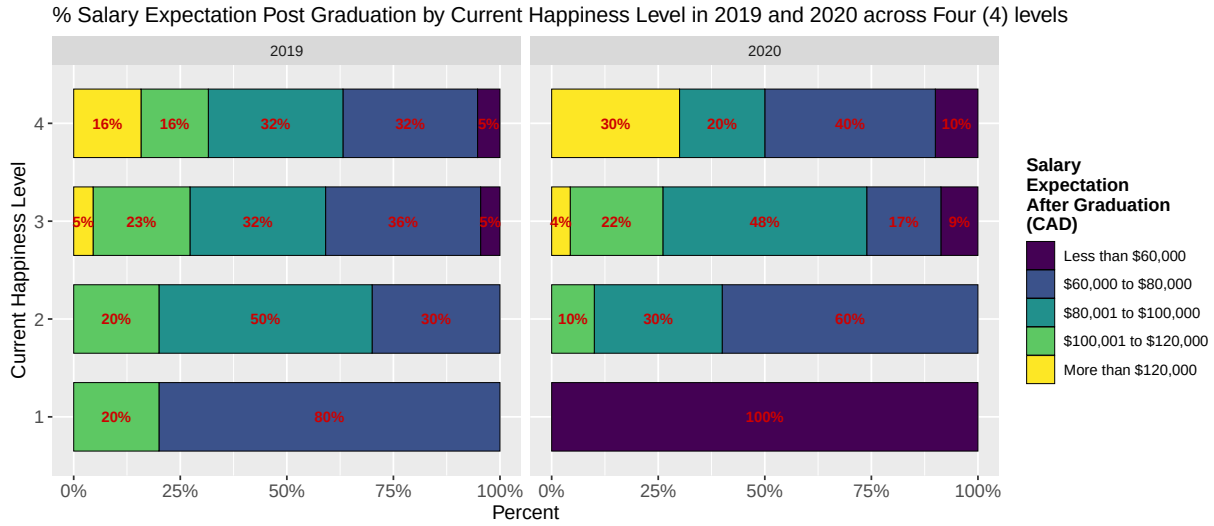
```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "currently_job_searching",
  "Currently Applying for \n Data Science Jobs", "% Salary Expectation Post Graduation by Currently Appl
)
```




```
# END SOLUTION
```

Percentage of `salary_exp_post_grad` vs `baseline_happiness` It is pretty noticeable that a higher happiness level is also associated with higher salary expectations. This trend applies for both years.

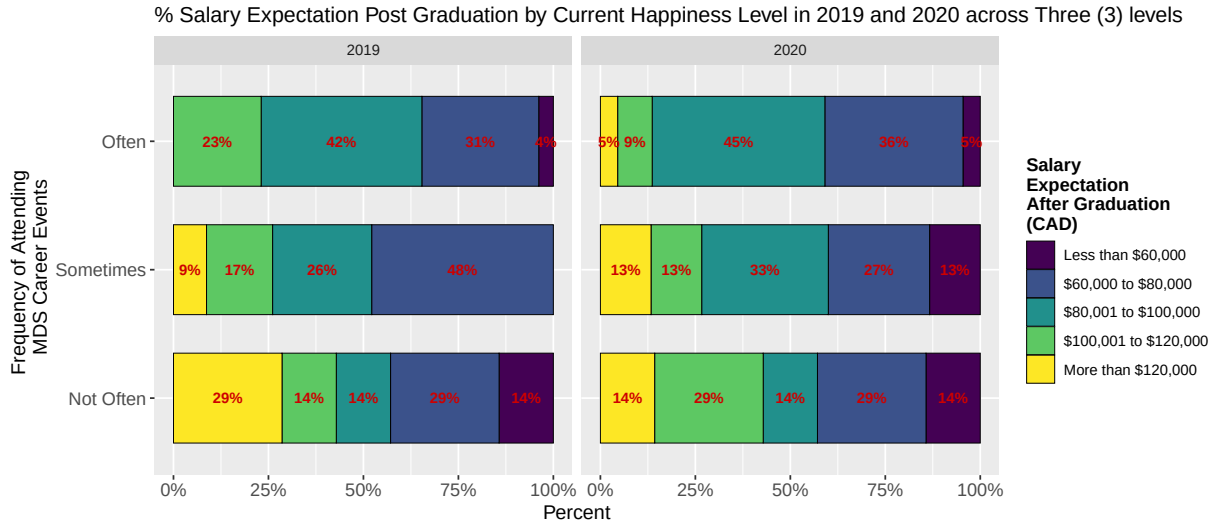
```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "baseline_happiness",
  "Current Happiness Level", "% Salary Expectation Post Graduation by Current Happiness Level in 2019 a
)
```




```
# END SOLUTION
```

Percentage of `salary_exp_post_grad` vs `freq_attend_mds_career_events` Those who do not often attend MDS career events also expect higher salaries after graduation.

```
# BEGIN SOLUTION
plotting_function(
  salary_df, "salary_exp_post_grad", "freq_attend_mds_career_events",
  "Frequency of Attending \n MDS Career Events", "% Salary Expectation Post Graduation by Current Happi

)
```




```
# END SOLUTION
```

Regression Model

Given that `salary_exp_post_grad` is an ordinal response, the most suitable regression model is the ordinal logistic regression. To fit the model with the package `MASS`, we use the function `polr()`, which obtains the corresponding estimates.

Let us begin by just using the X (`mds_self_rated_enjoy`) and Y (`salary_exp_post_grad`) of interest in

this observational study to estimate a regression model with these two variables.

However, before starting with the model fitting, it is important to highlight something regarding `mds_self_rated_enjoy` and the rest of the **non-binary** survey questions . These variables are ordinal. Thus, when fitting a regression with them as explanatory variables (regardless of whether the regression is classical, binomial logistic, count-type, etc.), we are likely interested in assessing the statistical significance of the difference between their ordered levels along with the corresponding interpretation. This will take us to the concept of contrasts.

```
options(contrasts = c("contr.treatment", "contr.sdif"))
```

Without Confounders Calculating the p-values for the regression coefficients along with the adjusted p-values via the Benjamini-Hochberg procedure with $\alpha = 0.05$ and Showing the model's summary via the function `tidy()`.

```
initial_model <- polr(salary_exp_post_grad ~ mds_self_rated_enjoy, Hess = TRUE, data = salary_df)
summary_initial_model <- tidy(initial_model) %>%
  mutate(p_value = pnorm(abs(statistic), lower.tail = FALSE) * 2) %>%
  mutate(p_value_adj = p.adjust(p_value, method = "BH")) %>%
  mutate_if(is.numeric, round, 2)


print(summary_initial_model)
```

```
## # A tibble: 7 x 7
##   term                estimate std.error statistic coef.type p_value p_value_adj
##   <chr>                  <dbl>     <dbl>     <dbl> <chr>       <dbl>       <dbl>
## 1 mds_self_rated_enj~     2.3       1.16      1.98 coeffici~    0.05        0.08
## 2 mds_self_rated_enj~    -0.68      0.75     -0.91 coeffici~    0.36        0.5
## 3 mds_self_rated_enj~    -0.33      0.42     -0.79 coeffici~    0.43        0.5
## 4 Less than $60,000|~    -2.66      0.47     -5.68 scale        0           0
## 5 $60,000 to $80,000~    -0.17      0.32     -0.53 scale        0.6         0.6
## 6 $80,001 to $100,00~     1.34      0.34      3.89 scale        0           0
## 7 $100,001 to $120,0~     2.69      0.45      6.01 scale        0           0
```

```
summary_initial_model[, c(1,7)] %>%
  filter(p_value_adj < 0.05)
```

```
## # A tibble: 3 x 2
##   term                                          p_value_adj
##   <chr>                                               <dbl>
## 1 Less than $60,000|$60,000 to $80,000                    0
## 2 $80,001 to $100,000|$100,001 to $120,000                0
## 3 $100,001 to $120,000|More than $120,000                 0
```

There is no significant effect of self-rated enjoyment of MDS on salary expectation after MDS graduation in any of the **successive differences** between the ordered levels in the factor self-rated enjoyment of MDS. However, three out of the four intercepts are significant. This indicates some differences in the ordered categories in the response variable, salary expectation after MDS graduation, which would be due to the different counts of each ordered category.

At this point, given that we have not analyzed confounders, we do not know if this null effect of self-rated enjoyment of MDS on salary expectation after MDS graduation is due to there truly being no effect of self-rated enjoyment of MDS, or if a confounder is masking this effect. To determine this, we need to investigate the effects of all possible confounders we can.

With Confounders Estimating the regression model called `full_model` of `salary_exp_post_grad` versus `mds_self_rated_enjoy` **along with the rest of the survey questions and year as STANDALONE confounders**. Calculating the p-values for the regression coefficients along with the adjusted p-values via the Benjamini-Hochberg procedure with $\alpha = 0.05$ and showing the model's summary via the function tidy().

```
full_model <- polr(salary_exp_post_grad ~ ., Hess = TRUE, data = salary_df)
summary_full_model <- tidy(full_model) %>%
  mutate(p_value = pnorm(abs(statistic), lower.tail = FALSE) * 2) %>%
  mutate(p_value_adj = p.adjust(p_value, method = "BH")) %>%
  mutate_if(is.numeric, round, 2)

print(summary_full_model)
```

```
## # A tibble: 26 x 7
##    term               estimate std.error statistic coef.type p_value p_value_adj
##    <chr>                 <dbl>     <dbl>     <dbl> <chr>       <dbl>       <dbl>
##  1 mds_self_rated_en~     3.36      1.59      2.11 coeffici~    0.04        0.11
##  2 mds_self_rated_en~    -2.94      1.01     -2.9  coeffici~    0           0.02
##  3 mds_self_rated_en~    -0.53      0.62     -0.86 coeffici~    0.39        0.63
##  4 salary_pre_mds$60~     1.16      0.56      2.08 coeffici~    0.04        0.11
##  5 salary_pre_mds$80~     2.66      0.92      2.89 coeffici~    0           0.02
##  6 salary_pre_mds$10~    -0.22      1.04     -0.21 coeffici~    0.83        0.85
##  7 salary_pre_mdsMor~     4.45      1.6       2.79 coeffici~    0.01        0.02
##  8 work_exp1 - 4 Yea~     1.58      0.58      2.7  coeffici~    0.01        0.03
##  9 work_exp4 - 7 Yea~    -0.34      0.62     -0.55 coeffici~    0.58        0.75
## 10 work_exp7 - 10 Ye~    -0.61      0.9      -0.68 coeffici~    0.5         0.68
## # ... with 16 more rows
```

```
summary_full_model[, c(1,7)] %>%
              filter(p_value_adj < 0.05)
```

```
## # A tibble: 7 x 2
##   term                                                          p_value_adj
##   <chr>                                                               <dbl>
## 1 mds_self_rated_enjoy3-2                                              0.02
## 2 salary_pre_mds$80,001 to $100,000-$60,000 to $80,000                 0.02
## 3 salary_pre_mdsMore than $120,000-$100,001 to $120,000                0.02
## 4 work_exp1 - 4 Years-Less than 1 Year                                 0.03
## 5 Less than $60,000|$60,000 to $80,000                                 0
## 6 $60,000 to $80,000|$80,001 to $100,000                               0
## 7 $100,001 to $120,000|More than $120,000                              0
```

Now, once we incorporate the confounders, we can see that `mds_self_rated_enjoy3-2` is significant on the response. Therefore, the confounders were masking this effect. Moreover, salary before starting MDS and years of previous work experience are also the only confounders that bear any statistical evidence that might be associated with salary expectation after MDS graduation.

In particular, `salary_pre_mds$80,001 to $100,000-$60,000 to $80,000` and `salary_pre_mdsMore than $120,000-$100,001 to $120,000`, and `work_exp1 - 4 Years-Less than 1 Year` have adjusted p-values of less than the significance level α (with 5% False discovery rate adjustment using the Benjamini-Hochberg procedure).

Is there a statistical difference between the data of both years?

Given the adjusted p-value for `year` in the summary of `full_model`, we can conclude that making a distinction between both years is not statistically significant.

Does the model without confounding variables perform better than a model with confounding variables

It is necessary to statistically check whether `full_model` provides a better data fit than the `initial_model`. Using $\alpha = 0.05$, conduct the corresponding hypothesis testing. Do not forget to specify the corresponding hypotheses.

The hypotheses are:

$$H_0 : \texttt{initial\_model} \text{ fits the data as good as the } \texttt{full\_model}$$

$$H_a : \text{otherwise}$$

Given the p-value obtained in the Likelihood Ratio Test (LRT), we can conclude that the `full_model` fits the data better than the `initial_model`.

```
anova(full_model, initial_model) %>%
   mutate_if(is.numeric, round, 2)
```

```
##
## 1
## 2 mds_self_rated_enjoy + salary_pre_mds + work_exp + ds_skill_confidence + does_optional_qs + curren
##   Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
## 1        93     276.63
## 2        74     198.96 1 vs 2    19    77.67       0
```

Reduced Model

Using Standalone explanatory varaibles t hatturned out to have at least one significant regression coefficient from the previous analysis. The variables are: `salary_pre_mds` and `work_exp`

Regarding our X of interest, we can see that `mds_self_rated_enjoy2-1` and `mds_self_rated_enjoy3-2` are statistically significant in this model. Moreover, in terms of the confounders used in this model, `salary_pre_mds$60,000 to $80,000-Less than $60,000`, `salary_pre_mds$80,001 to $100,000-$60,000 to $80,000`, `salary_pre_mdsMore than $120,000-$100,001 to $120,000`, and `work_exp1 - 4 Years-Less than 1 Year` are significant.

```
reduced_model <- polr(salary_exp_post_grad ~ mds_self_rated_enjoy +
  salary_pre_mds + work_exp,
Hess = TRUE, data = salary_df
)
summary_reduced_model <- tidy(reduced_model) %>%
  mutate(p_value = pnorm(abs(statistic), lower.tail = FALSE) * 2) %>%
  mutate(p_value_adj = p.adjust(p_value, method = "BH")) %>%
  mutate_if(is.numeric, round, 2)


summary_reduced_model[, c(1, 7)] %>%
  filter(p_value_adj < 0.05)
```

```
## # A tibble: 9 x 2
##   term                                                p_value_adj
##   <chr>                                                     <dbl>
## 1 mds_self_rated_enjoy2-1                                    0.01
## 2 mds_self_rated_enjoy3-2                                    0.02
## 3 salary_pre_mds$60,000 to $80,000-Less than $60,000         0.02
```

```
## 4 salary_pre_mds$80,001 to $100,000-$60,000 to $80,000         0.02
## 5 salary_pre_mdsMore than $120,000-$100,001 to $120,000        0.01
## 6 work_exp1 - 4 Years-Less than 1 Year                         0.02
## 7 Less than $60,000|$60,000 to $80,000                         0
## 8 $60,000 to $80,000|$80,001 to $100,000                       0
## 9 $100,001 to $120,000|More than $120,000                      0
```

Pairwise Comparison Between Three Models

Making pairwise comparisons between `initial_model`, `full_model`, and `reduced_model`.

Based on the testing results, we can conclude that we can use the `reduced_model`, since it is as good as the `full_model`), which fits the data better than the `initial_model`.

```
# BEGIN SOLUTION

LRT_full_vs_initial <- anova(full_model, initial_model) %>%
  mutate_if(is.numeric, round, 3)

print(LRT_full_vs_initial)
```

```
##
## 1
## 2 mds_self_rated_enjoy + salary_pre_mds + work_exp + ds_skill_confidence + does_optional_qs + curren
##   Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
## 1        93    276.628
## 2        74    198.961 1 vs 2    19   77.667       0
```

```
LRT_full_vs_reduced <- anova(full_model, reduced_model) %>%
  mutate_if(is.numeric, round, 3)

print(LRT_full_vs_reduced)
```

```
##
## 1
## 2 mds_self_rated_enjoy + salary_pre_mds + work_exp + ds_skill_confidence + does_optional_qs + curren
##   Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
## 1        85    209.026
## 2        74    198.961 1 vs 2    11   10.065   0.525
```

```
LRT_reduced_initial <- anova(reduced_model, initial_model) %>%
  mutate_if(is.numeric, round, 3)

print(LRT_reduced_initial)
```

```
##                                             Model Resid. df Resid. Dev   Test
## 1                              mds_self_rated_enjoy        93    276.628
## 2 mds_self_rated_enjoy + salary_pre_mds + work_exp        85    209.026 1 vs 2
##      Df LR stat. Pr(Chi)
## 1
## 2     8   67.602       0
```

```
raw_p_values <- c(
  LRT_full_vs_initial$`Pr(Chi)`[2],
  LRT_full_vs_reduced$`Pr(Chi)`[2],
  LRT_reduced_initial$`Pr(Chi)`[2]
)
```

```r
# Adjusting for multiple comparisons

test_results <- tibble(
  Test = c(
    "LRT Full Model vs. Initial Model",
    "LRT Full Model vs. Reduced Model",
    "LRT Reduce Model vs. Initial Model"
  ),
  raw_p_values
)
test_results <- test_results %>%
  mutate(p_value_adj = p.adjust(raw_p_values, method = "BH")) %>%
  mutate_if(is.numeric, round, 3)

print(test_results)
```

```
## # A tibble: 3 x 3
##   Test                                raw_p_values p_value_adj
##   <chr>                                      <dbl>       <dbl>
## 1 LRT Full Model vs. Initial Model           0           0
## 2 LRT Full Model vs. Reduced Model           0.525       0.525
## 3 LRT Reduce Model vs. Initial Model         0           0
```

Conclusion

Firstly, running a baseline model of the salary expectation after graduation versus the self-rated enjoyment of MDS does not provide any statistical evidence of an association between both variables. However, when adding all the potential confounders from the survey, we see that some of the coefficients for the association of self-rated enjoyment of MDS are now significant, even after adjusting for multiple comparisons in the regression coefficients.

Specifically, when we include all the potential confounders, there is evidence that there is a relationship between self-rated enjoyment of MDS on salary expectation after MDS graduation when increasing the self-rated score from 2 to 3. By conducting a Likelihood Ratio Test of the baseline versus this model, including all confounders, we have statistical evidence that taking into account these confounders makes the model fit the data better. Note that the confounders salary before joining MDS and previous work experience also show statistical evidence of association with the salary expectation.

To see if we could fit a better model, we refit it using salary before starting MDS and previous work experience as the only confounders. When we do this, the model is not significantly different from the model that includes all confounders. Thus, we choose this simpler model as our final model. When evaluating this final model's coefficients, there is specific evidence that there is a relationship between the expected salary when increasing the self-rated score from 1 to 2 and 2 to 3. This is true only when controlling for salary's confounding effect before starting MDS and previous work experience. Hence, it appears that salary before starting MDS and previous work experience mask the effect of self-rated enjoyment of MDS on salary expectation after MDS graduation (i.e., it is only observable when we take salary before starting MDS and work experience into account in the model). Note that the year where the survey was conducted does not have a statistical association with the response.

Is this relationship causal?

This is an observational study, so we cannot conclusively say, but we can at least assess the things we would need to assume to draw causal conclusions from an observational study and temper our causal claims based on that. The first assumption is that we have identified all confounders. We cannot prove this entirely.

However, we did assess many confounders, and only two of the ones we assessed were significant. So we have some slight evidence towards this assumption.

There might be something else that causes the variability observed in salary expectation after MDS graduation. However, we do not currently know what it is. If we were interested in further investigating this, it would be best to perform an additional observational study (this is not a study we could experiment for). We would test whether another variable is causal and again try to control for all possible confounders. Given what we learned about salary before starting MDS and salary expectation after MDS graduation, along with previous work experience, we would likely want to include these two variables in our new analysis as confounders.

Possible Improvement

Possible improvements to this analysis includes:

- A larger sample size.
- Asking for survey respondents to answer salary and year-related questions via inputting a number instead of a drop-down menu. This would allow the analysis to be done using linear regression (more straightforward to interpret and potentially more powerful).
- Alternatively, perhaps different boundaries could be chosen if they really want to stick with ordered categories. We could use what we learned from this study to pick better category boundaries for the future.
- Perform a cohort study so that some covariate questions could be asked at a more appropriate time (e.g., how confident in your data science skill set did you feel when first starting MDS on a scale of 1 - 4?) and the explanatory variable of interest (perceived enjoyment in MDS could be asked after completion of MDS, i.e., after the Capstone project experience).
- The final model assumes that the (X, Y) association is the same across strata (i.e., there are no interactions). This assumption might not hold. We can add an interaction term between the self-rated enjoyment of MDS and salary before starting MDS and previous work experience. If we see significant interaction coefficients, this would suggest that the (X, Y) association is not the same across strata.

Attribution

The question, data, and analysis that makes up this causal analysis were derived from a survey and analysis performed by the following past MDS students:

- Carrie Cheung.
- Alex Pak.
- Talha Siddiqui.
- Evan Yathon.