Predicting Data Science and STEM Salaries with Permutation Tests and Bootstrapping

5pm Section, Group 1: Robin Lee (robinlee), Eric Song (weichen_song), Justin Liu (justintliu)

June 07, 2022

Contents

Abstract

Uncertain of our future as post-undergraduates, we hope to utilize this analysis to gain a deeper understanding into the potential pecuniary benefits of obtaining a higher education, and whether sacrificing years of potential industry experience correlates with a better salary. In order to obtain a conclusive diagnosis, we will implement various bootstrapping techniques and permutation tests. We found that education level and years of experience were both significant predictors of total yearly compensation, but did not fit a linear regression model very well. Our best model, which contained an interaction between these two predictors, had a very low adjusted R^2 value of 0.1836, suggesting that only using one's education level and years of work experience to predict one's annual compensation in world's top companies is not a feasible approach.

Introduction

As students studying towards our Bachelor's degrees at UC Santa Barbara, we are naturally interested in whether or not spending the supplementary years obtaining a higher education—specifically a Master's Degree—following our tenure as undergraduates is worth the time and dedication compared to immediately entering the industry. Unwilling to waste time upon receiving our degrees and seeking the most profitable or beneficial option, it is often quite difficult for students, like us, to confidently come to a conclusion. In order to thoroughly measure exactly how much we are able to benefit from continuing our education for several more years, we can juxtapose individual salaries at each education level, while accounting for the number of years of experience in the industry, particularly in data science and STEM. Thus, we also hope to perceive whether completing a higher education is equivalent to the time gaining valuable industry experience.

Methods

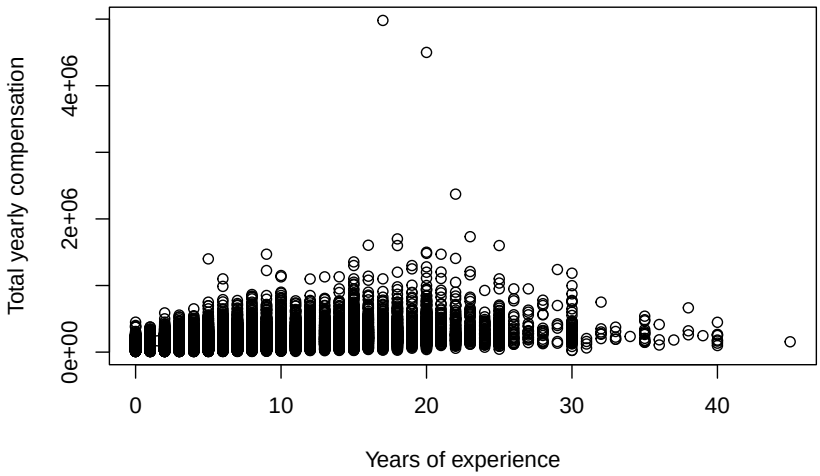
In this section, we will describe how we designed a simulation to answer whether higher education should be pursued over entering the industry. We also explain the intuition behind the R code we implemented to generate our analysis and results.

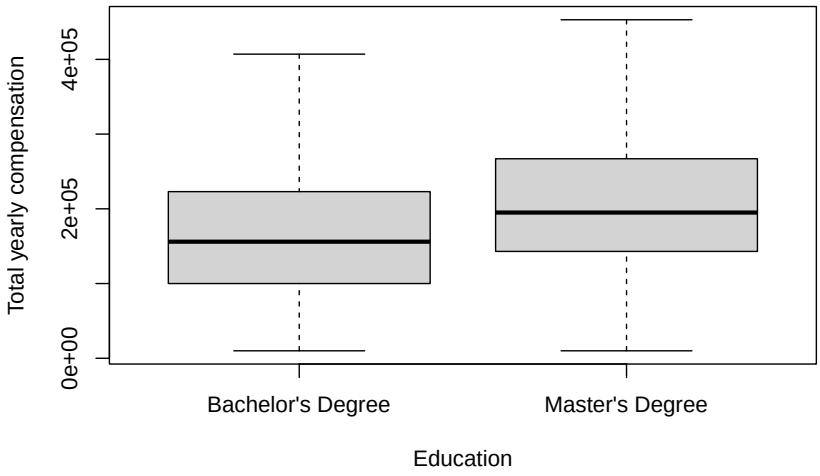
Data

The dataset we are using for this analysis (which was scraped off of levels.fyi and retrieved from Kaggle) contains information from over 62,000 individuals working in data science and STEM. With more than 20 variables and characteristics, the variables that are relevant to our particular study are:

Variable	Type
`totalyearlycompensation`	Numeric
`yearsofexperience`	Numeric
`Education`	String

Our response variable here is the total yearly compensation of an individual. We will focus on two predictors in this study: education level (binary, either Bachelor's or Master's) and years of experience. We present some of the exploratory analysis that we did on the data.



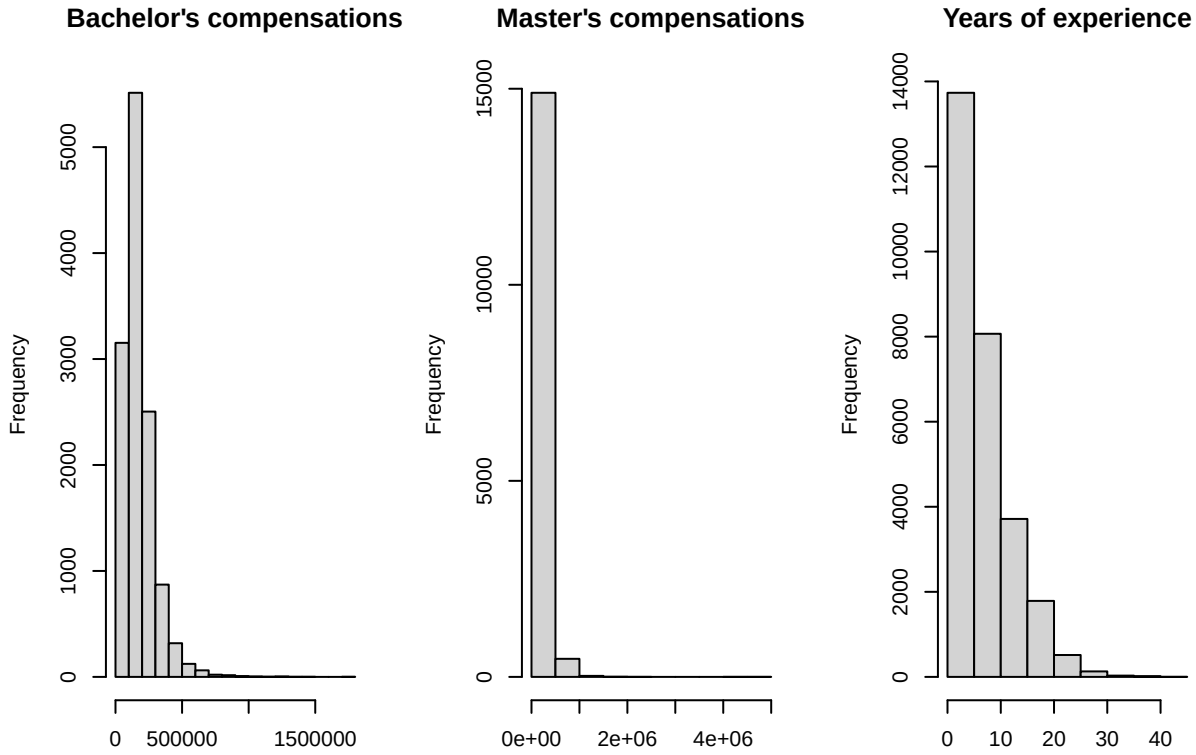


From the scatter plot, it is difficult to tell whether there is a correlation between total years of experience and total yearly compensation. This is due to the outliers that stretch the y-axis, making the distribution of points appear "flatter" than they really are. As for the boxplot, it appears that pursuing higher levels of education is positively correlated with total yearly compensation, though it is uncertain whether these differences between the levels are actually significant.

Algorithm and Model

Our goal is to estimate the salaries for each individual, given their education level and years of industry experience. Although our dataset contains a wealth of data, it is only a subset of the wider population of data science and STEM careers. Since there may be uncertainty when it comes to estimating the true parameters of the population, we decided to implement two methods—permutation tests and bootstrapping—in order to make inferences about our data.

Before using these tests, we tried the parametric versions of these tests, namely the two-sample t-test and Pearson's correlation test. The assumptions for the two-sample t-test is that the distribution of each group has to be normally distributed and have equal variabces. For the Pearson's correlation test, we need to check for the normality of the data. However, running our own analyses showed that these assumptions were not met. The distribution of compensations for bachelors, masters, and years of experience are not normal. Running Levene's test also suggested that the variance of the compensations among people with Bachelor's degrees and Master's degrees was unequal ($p = 1.969 \times 05$).



Permutation test is a method to test whether two groups come from the same distribution, given a numeric response. More specifically, for a two-sided permutation test, we are testing whether there is a significant difference in the mean numeric response between the two groups (i.e. one group could have a higher or lower mean than the other). The hypotheses are as follows:

- H_0: There is no difference between the distribution of the numeric responses for both groups. Any deivations are due to chance.
- H_A: There is a significant difference between the distribution of the numeric responses for both groups.

As the first step of the algorithm, the labels of the group are shuffled without replacement. We are only reordering the labels (hence the name *permutation* test) to ensure that the original count of the labels remains the same. This allows us to compare our simulated mean differences to the observed mean difference since we want to see if the groups come from the same distribution. Upon shuffling the labels, we compute the difference between the means of the new groups. After repeating this for many iterations, the simulated means begin to follow a curved distribution (for a two-sided test, the distribution decreases as the absolute mean differences get larger). We then find the proportion of simulations that are larger than our test statistic, which is the difference between the means of the two groups. This proportion is our p-value, the chance that the test statistic in the null model is equal to the observed values in the simulated values or further in the direction of the alternative. As a standard, p-values less than 0.05 means we reject the null hypothesis.

Since there are several binary predictors in our model, we thought running permutation tests on these variables would allow us to see whether or not having a degree beyond a Bachelor's would have an effect on yearly earnings. With the algorithm above, we can simulate many absolute mean differences under the null hypothesis and see if the actual difference is rare or not.

For our second method, we will utilize bootstrapping to find a suitable interval of estimates for the correlation between each numerical explanatory variable and the total yearly compensation (which is also numeric). For each level of education, we can construct confidence intervals to find a range of plausible estimates for the correlation between years of experience and total annual compensation. In order to implement this technique, we randomly draw individuals *with replacement* from the original dataset, allowing us to generate more samples from the population that are different from the original. We are able to perform this technique due to the Law of Averages, which states that the empirical probability of an event will converge towards the theoretical probability as more trials are repeated. Bootstrapping more samples makes the confidence intervals much narrower, allowing us to be fairly confident that the distribution of the correlation coefficients will be close to the true correlation. After generating many samples, we can aggregate all of the correlations and create a 95% confidence interval by taking the middle 95% of the data. This is done by marking the 2.5% quantile of the simulated statistics as the lower bound and the 97.5% quantile as the upper bound. If this confidence interval does not contain 0, then we can be fairly confident that there is some linear relationship between the numeric predictor and the total yearly compensation.

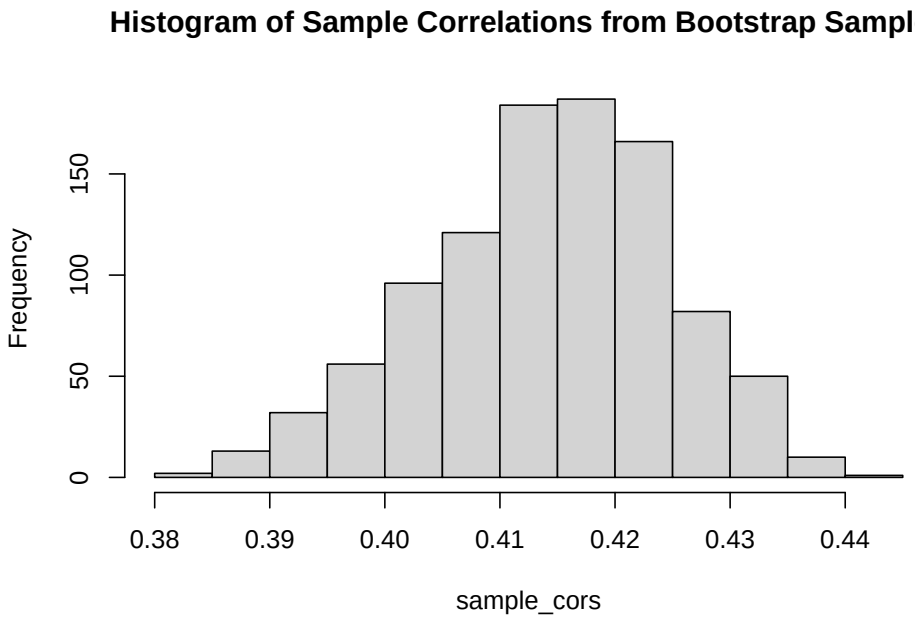
For linear regression, we decided to use simple and multiple linear regression models with total yearly compensation as our response variable (i.e. `y = totalyearlycompensation`). We ultimately implemented 4 models with different combinations of predictor variables: education (`y ~ Education`), years of experience (`y ~ yearsofexperience`), both education and years of experience (`y ~ Education + yearsofexperience`), and finally, the interaction (`y ~ Education * yearsofexperience`). In the results, we will discuss the Adjusted R^2 values in each of the aforementioned models, and conclude which model best explains the variability in total yearly compensation.

The overall goal of using these methods is to determine statistically significant variables that can be useful in predicting one's annual earnings in the STEM industry. As a brief summary, the permutation tests will be used to test whether the levels of each binary categorical variable differ significantly in their mean earnings. For numeric variables, we will test their relationship with annual earnings to see if their correlations are significantly different from 0. Finally, we will create a linear regression model to see if we are able to accurately predict one's annual compensations with one's education level and years of work experience.

Results

We can run permutation tests to see if the each binary variable could potentially explain some difference in mean earnings. We created our own function that takes in a number of iterations, a dataframe, a response variable, and a binary variable and runs a permutation test. From running the code with $n = 1000$ times, we found that there was a significant difference in the mean compensation between each level of education ($p = 0$ for all of the groups, see Appendix for the code and more detailed results). Additionally, the mean earnings appear to increase as the levels of education got higher, meaning that those with PhD's earned the most average, followed by those with Master's Degrees and then Bachelor's Degrees.

In the process of determining if there is a correlation between one's years of experience and total annual compensation, by bootstrapping our initial sample, we see that the histogram of correlation between variables `yearsofexperience` and `totalyearlycompensation` in each of our samples looks to be centered somewhere around 0.4.



The confidence interval for these correlations also don't contain 0 (see Appendix), which suggests that there is a somewhat weak positive correlation between years of experience and total annual compensation.

Using these metrics—education and years of experience—we fit several linear models to see if these predictors could possibly predict one's salary. We used model with just "Education" as the explanatory variable, model with only "yearsofexperience" as the explanatory variable, model with "Education + yearsofexperience", and model with the interaction between "Education and yearsofexperience". Although the predictors are highly significant ($p < 0.05$),the adjusted R^2 value of all the models are low, with the highest being 0.1836, which is the model of interaction between "Education" and "yearsofexperience". Since R^2 value is not a very deterministic indicator of the performance of a model, we also looked at AIC and BIC values of the models. Out of all of the models, the model with an interaction between the had the lowest AIC but the second-lowest BIC, just behind the model with the predictors without interactions. However, the AIC of the model without interactions was not that different from the model with interactions.

The results showed that the interaction model had the best performance among the models ($R^2 = 0.1836$). However, its still not a great model, implying that these two predictors alone may not be a good model to describe the data.

Discussion

We were able to show through different types of simulations (permutation tests and bootstrapping) that education level and years of experience were both significant predictors of total yearly compensation. However, fitting the linear regression model did not result in a good fit. Further studies may include accounting for more predictors to see if they can improve our model.

Given the results of the permutation tests, it makes sense those with more advanced degrees have higher salaries on average. They are usually have higher job positions and therefore higher paying jobs with more due to their expertise and esoteric knowledge. it also is not surprising that years of experience is positively correlated with salaries.

However, the data is not without limitations. A good portion of the data consists of observations that come from more well-known companies such as FANNG (Facebook, Apple, Amazon, Netflix, Google), which tend

to be higher-paying as well. This might be why some of the means were so high (over $200,000!) when focusing on the level of education. While this might be due to the accessibility of such data on a public website like levels.fyi, it nonetheless makes our averages much higher than expected.

So is it worth it to stay in school or use that time to acquire more industry experience? Both factors can lead to a higher average salary, so it's up to the individual to make that decision.

Appendix

Loading and Cleaning the Data

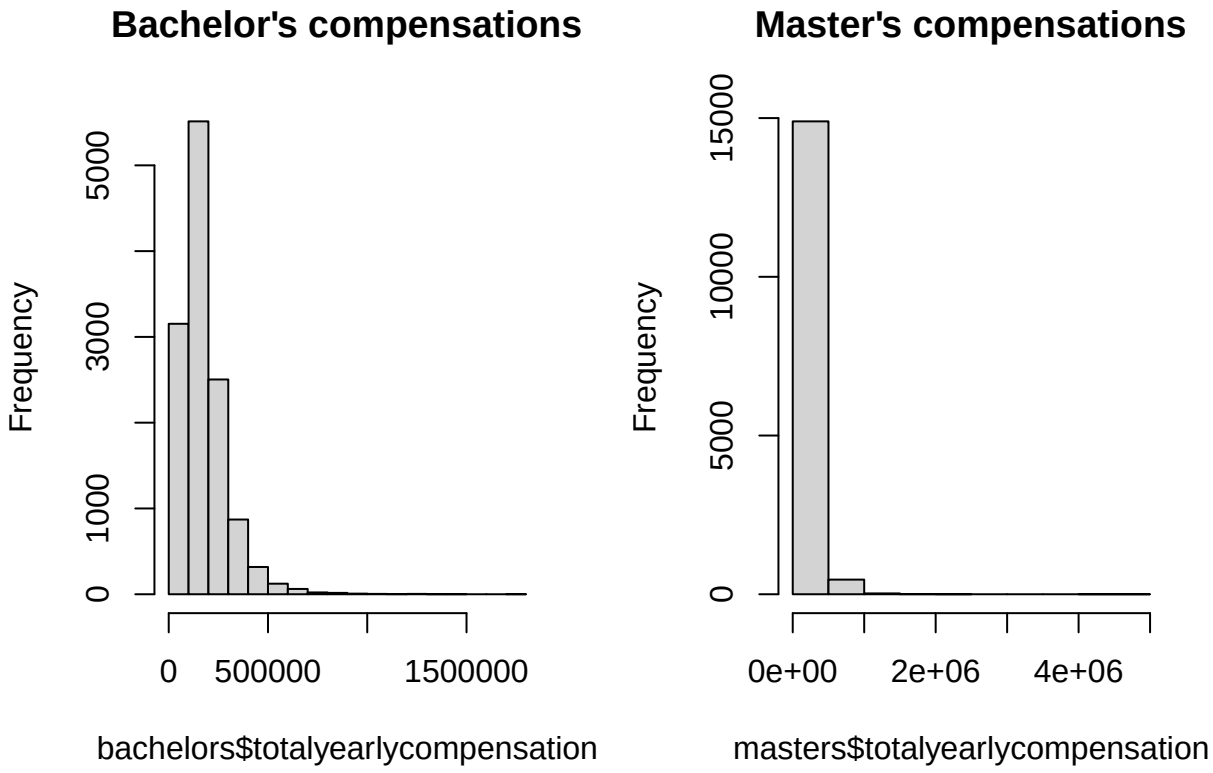
Exploratory Analysis

```
# scatterplot of compensation against experience
plot(totalyearlycompensation ~ yearsofexperience,
     data=data,
     xlab="Years of experience",
     ylab="Total yearly compensation")
```

```
# boxplot of compensation against education
boxplot(totalyearlycompensation ~ Education,
        data=data,
        outline=FALSE,
        ylab="Total yearly compensation")
```

Checking for Normality (t-test)

```
par(mfrow=c(1, 2)) # 1 row, 2 plots

bachelors <- subset(data, Education == "Bachelor's Degree") # people w/ bachelor's
masters <- subset(data, Education == "Master's Degree")     # people w/o master's

hist(bachelors$totalyearlycompensation, # histogram of bachelors' compensations
     main="Bachelor's compensations")
hist(masters$totalyearlycompensation,   # histogram of masters' compensations
     main="Master's compensations")
```



Checking for Equal Variance (t-test)

```
library(car)
```

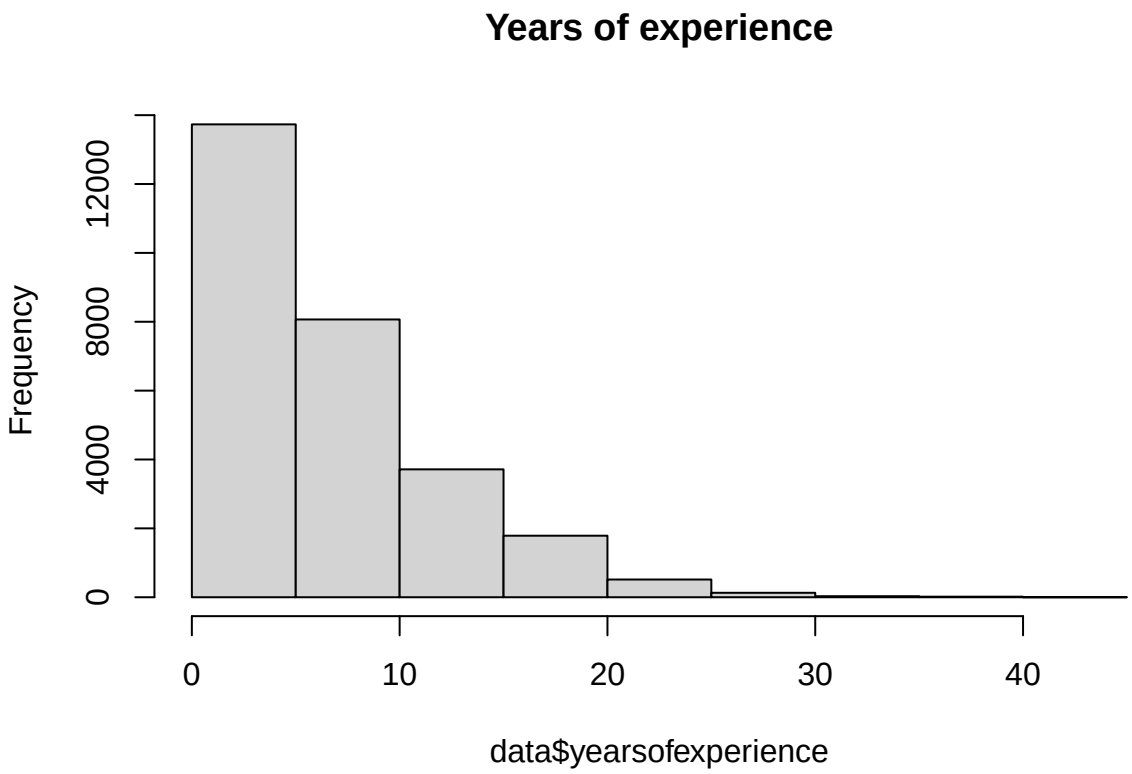
```
## Loading required package: carData
```

```
leveneTest(data$totalyearlycompensation ~ data$Education)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  18.225 1.969e-05 ***
##       27990
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Checking for Normality (Pearson's test)

```
hist(data$yearsofexperience,
     main="Years of experience")
```



Permutation Test

The arguments are defined as follows:

- `df` (dataframe): dataframe
- `response` (string): numeric variable in `df`
- `group` (string): binary variable in `df` (should be a factor), corresponding to values in `response`
- `column` (string): column in `df`
- `n` (integer): number of times to simulate the test

The function `perm_test()` performs a two-sided permutation test, which sees if two groups are significantly different in terms of a mean numeric response.

```
perm_test <- function(n, df, response, group) {
  # get the observed difference (the test statistic) and simulated differences
  obs_diff <- abs_mean_difference(df, response, group)
  sim_diffs <- n_mean_difference_sims(n, df, response, group)

  # calculate the observed mean difference and the two-sided p-value
  obs_mean <- tapply(df[, response], df[, group], mean)
  p.value <- sum(obs_diff <= sim_diffs) / n

  # print results
  cat(
```

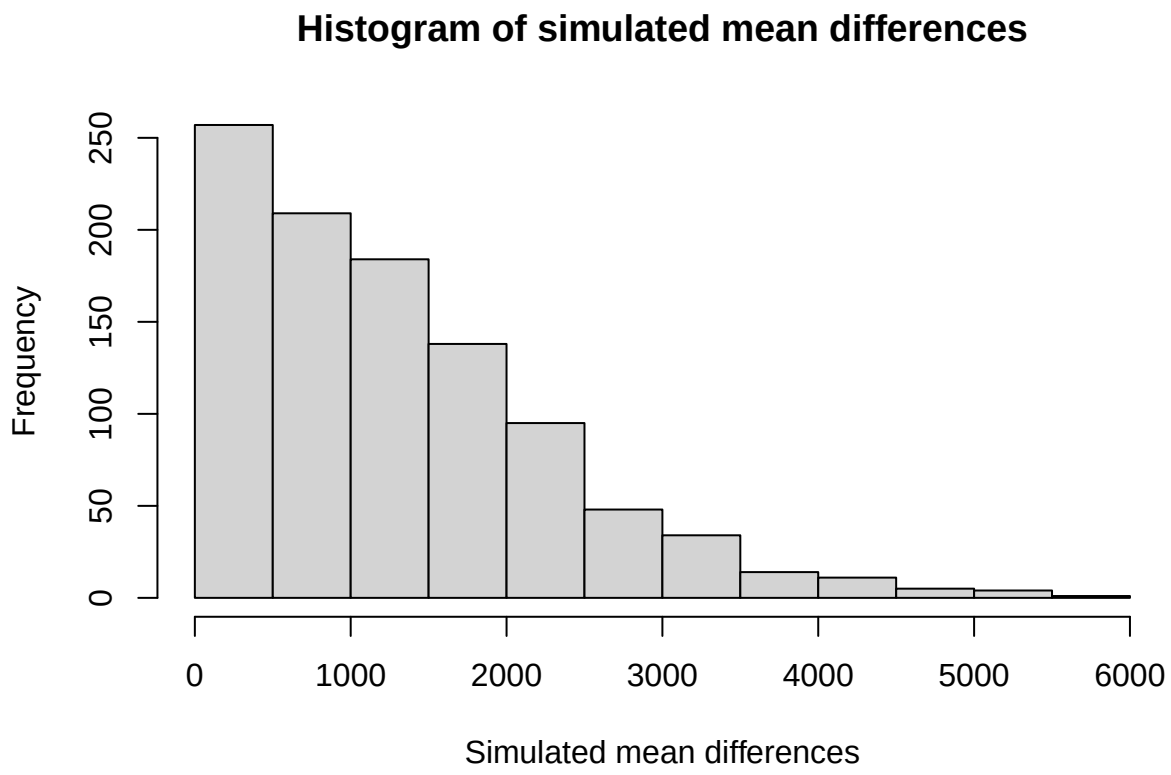
```
    # name of the group
    "\nName of factor:", deparse(substitute(group)),

    # 1st and 2nd levels of the group
    "\n1st level:", levels(df[, group])[1], "| mean =", obs_mean[1],
    "\n2nd level:", levels(df[, group])[2], "| mean =", obs_mean[2],

    # observed absolute mean difference and p-value from simulation
    "\n\nObserved absolute difference between the means of the levels:", obs_diff,
    "\np-value (two-sided) from simulating", n, "mean differences:", p.value)

  # print whether there is a significant difference (i.e. whether null hyp. is rejected)
  if (p.value <= 0.05) {
    cat("\n\nThere is a significant difference between the levels.")
  } else {
    print("\n\nThere is no significant difference between the levels.")
  }
}
```

Below are the helper functions used in `perm_test()`.

```
# returns the difference of the mean response between the 2 groups
abs_mean_difference <- function(df, response, group) {
  obs_mean <- tapply(df[, response], df[, group], mean)  # mean response for each group
  obs_diff <- abs(as.numeric(obs_mean[1] - obs_mean[2])) # absolute difference between means
  return (obs_diff)                                      # return absolute mean difference
}

# shuffles the order of values in a column
shuffle_labels <- function(df, column) {
  labels <- df[, column]    # subset the group from the dataframe
  num_obs <- length(labels) # number of observations

  shuffled_groups <- sample(labels, num_obs, replace=TRUE) # shuffle groups w/ replacement
  df$shuffled_groups <- shuffled_groups # add a new column to the dataframe w/ new groups

  return(df) # return updated dataframe
}

# simulates mean difference 1 time
mean_difference_sim <- function(df, response, group) {
  df <- shuffle_labels(df, group) # shuffle the groups

  # calculate the mean difference between the two new groups
  diff <- abs_mean_difference(df, response, "shuffled_groups")

  return (diff) # return mean difference
}

# simulates mean difference n times
n_mean_difference_sims <- function(n, df, response, group) {
  all_diffs <- c()                                    # holds all simulated differences

  for (i in 1:n) {                                    # for each iteration
```

```
    diff <- mean_difference_sim(df, response, group) # simulate mean difference
    all_diffs[i] <- diff                             # store in all_diffs
  }

  hist(all_diffs, breaks=20, # histogram of simulated differences
       main="Histogram of simulated mean differences",
       xlab="Simulated mean differences")
  return (all_diffs)         # return the simulated differences
}
```

```
perm_test(1000, data, "totalyearlycompensation", "Education")
```




```
##
## Name of factor: "Education"
## 1st level: Bachelor's Degree | mean = 177845.4
## 2nd level: Master's Degree | mean = 220731.1
##
## Observed absolute difference between the means of the levels: 42885.67
## p-value (two-sided) from simulating 1000 mean differences: 0
##
## There is a significant difference between the levels.
```

Bootstrapping Correlations

```
# Find the correlation between years at company and total annual compensation for our sample
cor.test(data$yearsofexperience,       # correlation between years of experience
         data$totalyearlycompensation, # and yearly compensation
         alternative = 'two.sided',    # two-sided test
         exact=FALSE,                  # avoid ties in data
         method = 'spearman')$estimate # get the Spearman's correlation estimate

# Obtain more samples from the population using bootstrapping
bootstrap_cor <- function() {
  re_sample <- data[sample(nrow(data), replace = TRUE),]
  cor <- cor.test(re_sample$yearsofexperience,
                  re_sample$totalyearlycompensation,
                  alternative = 'two.sided',
                  exact=FALSE,
                  method = 'spearman')$estimate
  return (cor)
}

# Do 1000 repetitions
set.seed(1)
n <- 1000
sample_cors <- c()
for (i in 1:n){
  sample_cors[i] <- bootstrap_cor()
}

# Plot a histogram to visualize
hist(sample_cors, main = "Histogram of Correlations from Bootstrap Samples")

# Construct a confidence interval for all sample correlations
CI_cor <- c(quantile(sample_cors, probs = 0.025), quantile(sample_cors, probs = 0.975))
CI_cor
```

Fitting the Linear Models

```
# education level as the only predictor
fit1 <- lm(totalyearlycompensation ~ Education, data=data)
summary(fit1)
```

```
##
## Call:
## lm(formula = totalyearlycompensation ~ Education, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -210731  -77731  -23845   45269 4759269
##
## Coefficients:
```

```
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 177845       1154  154.06   <2e-16 ***
## EducationMaster's Degree     42886       1557   27.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 129600 on 27990 degrees of freedom
## Multiple R-squared:  0.02639,    Adjusted R-squared:  0.02636
## F-statistic: 758.8 on 1 and 27990 DF,  p-value: < 2.2e-16
```

```
# years of experience as the only predictors
fit2 <- lm(totalyearlycompensation ~ yearsofexperience, data=data)
summary(fit2)
```

```
##
## Call:
## lm(formula = totalyearlycompensation ~ yearsofexperience, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -409838  -67908  -13191   47092 4686593
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       134174.8     1138.0  117.91   <2e-16 ***
## yearsofexperience   9366.6      123.3   75.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 119600 on 27990 degrees of freedom
## Multiple R-squared:  0.1709, Adjusted R-squared:  0.1708
## F-statistic:  5768 on 1 and 27990 DF,  p-value: < 2.2e-16
```

```
# education and years of experience as predictors
fit3 <- lm(totalyearlycompensation ~ Education + yearsofexperience, data=data)
summary(fit3)
```

```
##
## Call:
## lm(formula = totalyearlycompensation ~ Education + yearsofexperience,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -402262  -66922  -14090   45755 4676186
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              119947.1     1319.0   90.94   <2e-16 ***
## EducationMaster's Degree  29987.9     1436.5   20.88   <2e-16 ***
## yearsofexperience          9051.7      123.3   73.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 118700 on 27989 degrees of freedom
## Multiple R-squared:  0.1836, Adjusted R-squared:  0.1835
## F-statistic:  3147 on 2 and 27989 DF,  p-value: < 2.2e-16
```

```
# education, years of experience, and their interaction as predictors
fit4 <- lm(totalyearlycompensation ~ Education * yearsofexperience, data=data)
summary(fit4)
```

```
##
## Call:
## lm(formula = totalyearlycompensation ~ Education * yearsofexperience,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -411102  -66654  -13970   45654 4674004
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              121832.2     1584.4  76.893   <2e-16
## EducationMaster's Degree                  26243.2     2259.6  11.614   <2e-16
## yearsofexperience                          8757.0      184.5  47.459   <2e-16
## EducationMaster's Degree:yearsofexperience   532.5      248.0   2.147   0.0318
##
## (Intercept)                              ***
## EducationMaster's Degree                 ***
## yearsofexperience                        ***
## EducationMaster's Degree:yearsofexperience *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 118700 on 27988 degrees of freedom
## Multiple R-squared:  0.1837, Adjusted R-squared:  0.1836
## F-statistic:  2100 on 3 and 27988 DF,  p-value: < 2.2e-16
```

```
cbind(
  AIC=AIC(fit1, fit2, fit3, fit4)$AIC,
  BIC=BIC(fit1, fit2, fit3, fit4)$BIC
)
```

```
##           AIC      BIC
## [1,] 738492.0 738516.7
## [2,] 733995.8 734020.6
## [3,] 733565.3 733598.3
## [4,] 733562.7 733603.9
```

References

[1] https://www.kaggle.com/datasets/jackogozaly/data-science-and-stem-salaries

[2] https://inferentialthinking.com/chapters/11/3/Decisions_and_Uncertainty.html#the-p-value

[3] https://inferentialthinking.com/chapters/12/1/AB_Testing.html

[4] https://inferentialthinking.com/chapters/13/2/Bootstrap.html