# Statistical Inference of Salary Increase By Major

CS50: Formal Analysis

Minerva Schools at KGI

Nina Hamidli
December 2019

**Table of contents**

## 1. Introduction

Choosing a major in college is always a hard decision for the students. Based on the dataset given about Undergraduate Majors and their corresponding salaries, we will look into which set of majors is more profitable to go into. The original dataset about salaries and majors gives us information about median salaries right after graduation and in the mid-career of the graduates. To find out if getting a degree in Computational Sciences (CS) majors will be more or less profitable than getting a Social Sciences (SS) degree, we use statistical and practical significance tests. So, the question we will be discovering is "Based on the gathered data of current graduates, are the CS majors earning more than the SS graduates?". Analyzing this question will give us valuable information about which majors are better investments.

## 2. Dataset.

The dataset used in the analysis is from The Wall Street Journal (2017) dataset of *"Salary Increase By Major"*. The data is gathered from a year-long research involving 1.2 million people from over 300 US colleges with only a bachelor's degree by PayScale Inc. about their salaries. The dataset includes information about the change in the salary from the start to mid-career, percentiles of the mid-career salaries from 10th to 90th and median start and mid-career salaries. From this dataset, we will be working with two columns of the table comparing the salaries by majors, Undergraduate Major name and Mid-Career Median Salary, as we will be using the hypothesis testing for the difference of the mean. Those two columns will serve as our alleged independent and dependent variables, respectively. The former, being qualitative and categorical, will help us divide the dataset into different categories such as CS majors and SS majors, and the latter, which is a quantitative and continuous variable, but treated as discrete because it is

rounded in dataset, showing median salaries, will help us make numerical conclusions about the data, such as graduates with X degree earn XXX more on average than graduates with Y degree. The confounding variable could be the school from where the degree is obtained since this also contributes to the final results after graduation. Thus, serving as another independent variable the name of the school can have an effect on the salaries of the graduates. While we don't really measure if the worker has gotten a master's degree or not, the degree after undergraduate can still play a role in determining the salaries of workers. The master's degree ownership can serve as the extraneous variable in this scenario.

For the analysis, I have sorted out data into two categories: CS majors and SS majors median mid-career salaries. As I am aiming to help Minerva students in their decision about which major to declare next year, the categories include majors which are similar to the ones offered at Minerva.

## 3. Methods

The whole analysis and calculation process are conducted using various Python packages. Initially, we are reading the data using the Pandas library. Before starting to work with data, we examine it and convert all strings (values containing dollar signs) into integers (the code for this is shown in Appendix A), so we can process it further in our calculations. We also break the data into two lists: one with CS majors and another one with SS majors. Table 1 shows the summary of statistics for the median mid-career salary across given two categories of majors (all calculations are shown in Appendix B).

| | Major Type | Number of majors | Mean of salaries | Median of salaries | Standart Deviation |
|---|---|---|---|---|---|
| 0 | Computational Sciences | 10 | 89900.0 | 93550.0 | 13325.239210 |
| 1 | Social Sciences | 10 | 71310.0 | 68350.0 | 12637.361275 |

**Table 1.** Summary of descriptive statistics (number of entrants, mean and standard error of salaries) for the median mid-career salary for our two categories: CS and SS majors.

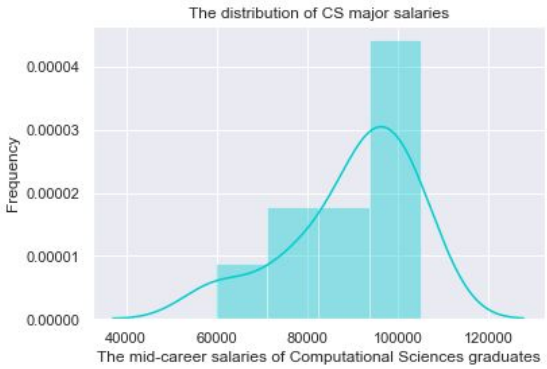



**Figure 1. The mid career salaries of CS majors.**

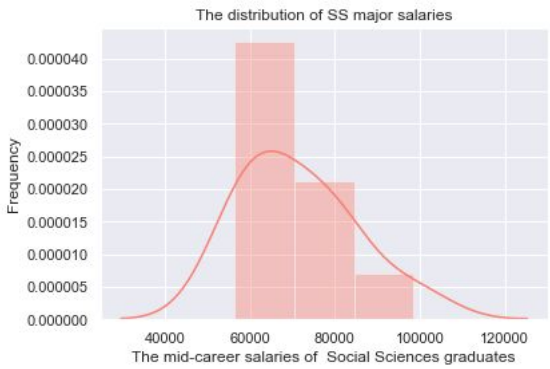



**Figure 2. The mid career salaries of SS majors.**

The figure 1&2 shows the distribution of salaries for CS and SS majors using Python seaborn library for making histograms. The code for this section is located in Appendix C.

From the histograms, we see that CS majors distribution (Figure 1) is somewhat normal with a slight skew to the right, while SS majors (Figure 2) distribution is skewed to the left. Another distinctive feature which we are observing is that most of the CS majors earn around 90000$, whereas SS around 70000$. The left-skewness of the SS salaries shows that the majority of SS degree holders earn in a range from 60000$ to 70000$. The mean of this dataset is 71310$, while the median is 68350$ from Table 1 which is also shown on the histogram as it is

left skewed because the median is less than the mean. We can observe the same trend also as the most frequency is concentrated between 60000$ and 70000$ marks. Those statistics indicate that among SS majors, the majority are more likely to earn more than 68350$. The frequency of the salaries decreases as the amount of salaries increases, indicating that the majority of SS graduates earn around the same range of salary, less 70000$ but more than 55000$.

According to the histogram, most salaries of CS majors get no less than 90000$ and no more than 11000$. In this case, the median is greater than the mean according to Table 1 which explains why the histogram is skewed to the right. This also indicates that the majority of CS majors are likely to get less than 93550$ on average.

Firstly, we are creating a 95% confidence interval of the difference between two means to find a range of possible values for an actual population mean. We can use the population standard deviation, to compute the standard error. But we don't really know the population standard deviation because we are analyzing samples. That's why we will be using an estimate of sample standard deviation and t-score instead of z-score since our population size is less than 30 and subsequently we don't satisfy the Central Limit Theorem.

In our calculation, the t-value will depend on the degrees of freedom (df = n-1) because of the small sample size. As T-distribution is adjusted for smaller sample size, if we increase the sample size, the t-distribution gets closer and closer to the normal distribution. So, we will be treating our distributions as normal. We will be using the formula for T-distribution for our confidence interval:

$$95\%\ \text{Confidence Interval} = \bar{x} \pm t * \frac{SD}{\sqrt{n}}$$

Where $\bar{x}$ is the sample means, t is t-score, SD - standard deviation and n is the sample size. From the calculations in Appendix D, we get that the confidence interval for CS majors is [76411$, 103389$] and for SS - [58518$, 84102$]. This tells us that the 95% confidence interval is within those ranges for each of the major types. Thus, we can be 95% confident that the true mean of the population median salaries by mid-career for CS and SS majors is within the given ranges.

To answer the question of the research, we are also setting two hypotheses. First one is a null hypothesis: Getting a CS or SS degree doesn't affect your mid-career salary. While our alternative hypothesis is that finishing a CS degree will lead to higher salaries than an SS degree. We will conduct the one-tail hypothesis testing, as our condition involves alternative testing for checking if one is greater than the other. Thus, we will be looking at differences only on one side by using calculated statistical information from Table 1.

In order to prove the existence of the relationship between our two variables not caused by chance, we need to use statistical significance. To do so, we will use hypothesis testing for the difference of means. All the calculations for this section of calculating statistical and practical significance are made in Python (shown in Appendix E). Step by step approach of finding the statistical significance would be:

1. Difference of the means: $\bar{x}_1 - \bar{x}_2 = 18590$
2. Significance level: $\alpha = 5\%$ or 0.05
3. Standard error: $SE = \sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}} = 5807.45$
4. Degrees of freedom: $n - 1 = 9$ *since we have same sample size for both of the samples*

Because the condition of the independence of the variable is met, since we divided the samples by the majors by assuming that the sample size for gathering this information was random and there is no relationship between types of majors, we can use the difference of the mean test to show statistical significance.

1. Null Hypothesis : $\bar{x}_2 - \bar{x}_1 = 0$
2. Alternative Hypothesis: $\bar{x}_1 - \bar{x}_2 > 0$
3. Significance level: $\alpha = 5\%$ or 0.05
4. T : $\frac{\bar{x}_1 - \bar{x}_2}{SE} = \frac{18590}{5807.45} = 3.2$
5. P-Value: 0.0054 => P-value < 0.05

From those calculations, we can reject the null hypothesis as P-value is immensely less than the significance level of 5% or 0.05. As we are rejecting H0, we could conduct Type 1 error if our H0 is actually true. This would lead to false thinking that no matter which major you choose, you will earn the same. However, this error is not what would concern as us people would still end up at places they want and be happy as money didn't play a role in their decision. For Type 2 error, we would come to the conclusion that there is a difference when there is actually none. This potentially could lead people to choose majors which they don't really want just for the sake of earning more and end up earning the same as others which would be very disappointing. Thus, for our analysis, Type 2 error should be avoided. This test showed us that there is a difference between the two but doesn't show how significant this difference is. Thus, we need to also calculate practical significance which will demonstrate the difference between the magnitude of

examined two groups. So, practical tests will describe how much more it is profitable to study CS majors rather than SS.

Out of all ways to calculate the effect size, we can use Cohen's D to find out the effect size since we have the same sample sizes (10) and similar standard deviations around 13000 . But will be using the Hedges G since our sample size is less than 30 to avoid upward bias because of the sample size. For this, we initially calculated the Cohen's D and then find Hedge's G. For Cohen's D the calculations are done based on this formula:

$$D = \frac{\bar{x}_1 - \bar{x}_2}{S_{pooled}}$$

1. The first step is to find the difference of the means which is 18590
2. For the second step, we need to calculate the Pooled Standard Deviation using

$$S_{pooled} = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}$$, which equals to 1.358.

From here, we now can use the refined formula of Cohen's D to find Hedge's G.

$$g = Cohen's\ d * (1 - \frac{3}{4(n_1 + n_2) - 9})$$

We get that practical significance is 1.30, which means that the effect size is of large significance as it is more than 0.8 according to Cohen's definitions of levels of significance. This means that the difference between our groups (CS and SS) is very substantial and the choice of CS majors will have a large positive effect on the mid-career salary.

## 4. Results and Conclusions

Throughout this paper, we have applied inductive reasoning to make inferences about the process that generated the data to generalize it to the whole population. We have analyzed using

statistical tools the sample data and drawn general conclusions based on the given information to show that there is a correlation between our variables : type of major and salaries in mid-career.

From our analysis, we saw that the decision between studying between CS and SS majors will affect the mid-career salary. We are 95% sure that the actual mean of the population for SS or CS salaries lies between the calculated ranges in Appendix D. The analysis also showed that we reject the Null Hypothesis since our P-value,0.0054, calculated in Appendix E, was significantly smaller than alpha value (0.05). This demonstrated that the salaries of SS and CS majors are dependent variables since as we change the type of the major the salary either increases or decreases. Then, the usage of both practical and statistical significance tests showed that there is substantial difference between salaries of people with CS and SS degrees. As our practical significance has a high effect, we can conclude that choosing a CS major is more profitable than an SS major.

# Appendix

The data can be accessed here.

## Appendix A: Importing and analyzing the data.

```
#importing useful libraries
import numpy as np
import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
from scipy import stats
from scipy.stats import sem, t
from scipy import mean


#reading datasets by majors which include starting, median, and percentile salaries
majors = pd.read_csv('Downloads/college-salaries/degrees-that-pay-back.csv')

#setting up dollar variable which contains only collumns involving dollar sign
dollar_sign = ['Starting Median Salary','Mid-Career Median Salary','Mid-Career 10th Percentile Salary',
               'Mid-Career 25th Percentile Salary','Mid-Career 75th Percentile Salary',
               'Mid-Career 90th Percentile Salary']

#converting string values of the majors dataset containing dollar signs into string values
for x in dollar_sign:
    majors[x] = majors[x].str.replace("$","")
    majors[x] = majors[x].str.replace(",","")
    majors[x] = pd.to_numeric(majors[x])

#show first 5 rows of data
majors.head()
```

| | Undergraduate Major | Starting Median Salary | Mid-Career Median Salary | Percent change from Starting to Mid-Career Salary | Mid-Career 10th Percentile Salary | Mid-Career 25th Percentile Salary | Mid-Career 75th Percentile Salary | Mid-Career 90th Percentile Salary |
|---|---|---|---|---|---|---|---|---|
| 0 | Accounting | 46000.0 | 77100.0 | 67.6 | 42200.0 | 56100.0 | 108000.0 | 152000.0 |
| 1 | Aerospace Engineering | 57700.0 | 101000.0 | 75.0 | 64300.0 | 82100.0 | 127000.0 | 161000.0 |
| 2 | Agriculture | 42600.0 | 71900.0 | 68.8 | 36300.0 | 52100.0 | 96300.0 | 150000.0 |
| 3 | Anthropology | 36800.0 | 61500.0 | 67.1 | 33800.0 | 45500.0 | 89300.0 | 138000.0 |
| 4 | Architecture | 41600.0 | 76800.0 | 84.6 | 50600.0 | 62200.0 | 97000.0 | 136000.0 |

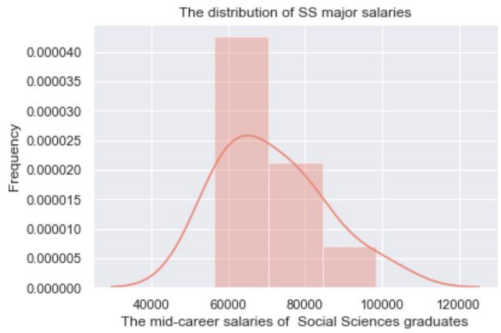
## Appendix B: Descriptive Statistics calculations.

```
#dividing majors into 2 dataset according to Minerva colleges
#computational sciences
cs = ["Computer Science" , "Math", "Information Technology (IT)", "Computer Engineering",
      "Management Information Systems (MIS)", "Graphic Design", "Industrial Engineering", "Civil Engineering",
     "Electrical Engineering", "Aerospace Engineering"]
#social sciences
ss =["Economics", "Political Science", "Journalism", "Psychology","Sociology", "Forestry", "Criminal Justice",
     "Communications", "Philosophy", "International Relations"]
#lists for the median salary of the two new datasets
#even though we don't have actual control/test groups as we don't do interventional study
#I'm naming new list test and control to make clear difference for myself later in code
test = []
control = []

#iteratting through rows of the original dataset to create two lists containing only mid career median salary
#for SS and CS majors
for index, row in majors.iterrows():
    if row['Undergraduate Major'] in cs:
        test.append(row["Mid-Career Median Salary"])
    elif row['Undergraduate Major'] in ss:
        control.append(row["Mid-Career Median Salary"])

#calculating mean, median and standart deviation of test and control datasets
#numpy mean function which takes in a given list and returns the average of the elements in given list of data
mean_cs = round(np.mean(test))
mean_ss = np.mean(control)
#numpy median function which takes in a given list and returns the median of the list elements
median_cs= np.median(test)
median_ss= np.median(control)
#numpy std function which takes in a given list and returns the standa rd deviation, the spread of a distribution,
#of the elements in a given list
sd_cs = np.std(test)
sd_ss = np.std(control)


#calculating the number of elements in each group
n_cs= len(test)
n_ss= len(control)

#using sem() function from scipy package, we calculate the standard error of the mean of the values in test and
#control lists by using formula of standart error of the mean(SEM) - s/sqrt(n) where s is the sample standard
#deviation and n is the sample size
std_err_cs = sem(test)
std_err_ss = sem(control)

class display(object):
    #Display HTML representation of multiple objects, the code for making a table is taken from
    #https://jakevdp.github.io/PythonDataScienceHandbook/03.07-merge-and-join.html
    template = """<div style="float: left; padding: 12px;">
    <p style='font-family:"Times New Romans", Courier, monospace'>{0}</p>{1}
    </div>"""
```

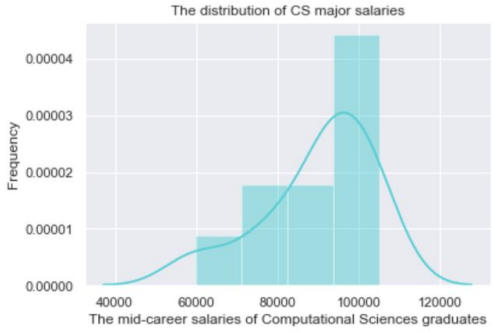
```
    def __init__(self, *args):
        self.args = args

    def _repr_html_(self):
        return '\n'.join(self.template.format(a, eval(a)._repr_html_())
                         for a in self.args)

    def __repr__(self):
        return '\n\n'.join(a + '\n' + repr(eval(a))
                           for a in self.args)

#creating a table using calculated stats
figure1 = pd.DataFrame({'Major Type': ['Computational Sciences', 'Social Sciences'],
                        'Number of majors': [n_cs, n_ss],
                        'Mean of salaries': [mean_cs, mean_ss],
                        'Median of salaries': [median_cs, median_ss],
                       'Standart Deviation':[sd_cs, sd_ss]})
display('figure1')
```

## Appendix C: Graphing histogram

```
#making histograms using seaborn library for cotrol group
sns.set()
#importing our test group data
sns.distplot(control, color = 'salmon')
#labeling
plt.title("The distribution of SS major salaries")
plt.xlabel("The mid-career salaries of  Social Sciences graduates")
plt.ylabel("Frequency")
plt.show()
```




```
#making histograms using seaborn library for test group
sns.set()
#importing our test group data
sns.distplot( test, color = 'darkturquoise' )
#labeling
plt.title("The distribution of CS major salaries")
plt.xlabel("The mid-career salaries of Computational Sciences graduates")
plt.ylabel("Frequency")
plt.show()
```




## Appendix D: Calculating Confidence Interval using t-distribution

```
#calculating confidence interval
confidence = 0.95
#by using confidence intervals formula for t-distribution, we calculate the multiplication of standart deviation/sqrt
#of a sample size by t-score for cs:
h = t_score * np.sqrt(sd_cs**2/n_cs)
#calculating lower bond
start_cs = mean_cs - h
#upper bond
end_cs = mean_cs + h
print("The lower bound of the interval for cs is " + str(round(start_cs)))
print("The upper bound of the interval for cs is " + str(round(end_cs)))

#by using confidence intervals formula, we get for ss:
h = t_score * np.sqrt(sd_ss**2/n_ss)
#calculating lower bond
start_ss = mean_ss - h
#upper bond
end_ss = mean_ss + h
print("The lower bound of the interval for ss is " + str(round(start_ss)))
print("The upper bound of the interval for ss is " + str(round(end_ss)))
```

```
The lower bound of the interval for cs is 76411.0
The upper bound of the interval for cs is 103389.0
The lower bound of the interval for ss is 58518.0
The upper bound of the interval for ss is 84102.0
```

## Appendix E: Calculating statistical and practical significance

```
#finding point estimate for my data samples
#we are substracting from mean_cs since our alternative hypothesis states that CS majors earn more
point_estimate = mean_cs - mean_ss
#calculating standart error of the sample population of CS and SS majors using
standard_error = np.sqrt(sd_cs**2/n_cs + sd_ss**2/n_ss)
#finding t-score by using point estimate and standart error
t_score = (point_estimate - 0)/standard_error
print ("Point estimate is", point_estimate)
print ("Standart error is",standard_error)
print ("T-score is",t_score)
```

```
Point estimate is 18590.0
Standart error is 5807.451248181082
T-score is 3.201060018510266
```

```
#creating a function which will calculate the difference of means. taking in both of the data subsets and the number
#of tails
def dmt(data1,data2,tails):

    #including Bessel's correction as our sample sizes are less than 30 to avoid conducting type 1 error to find
    #standart deviation for both samples
    #for cs
    s1 = np.std(data1,ddof=1)
    #for ss
    s2 = np.std(data2,ddof=1)

    #degrees of freedom
    df = min(n_cs,n_ss) - 1
    #calculating p-value using numbers of tails, t_score and degrees of freedom
    p_value = tails*stats.t.cdf(-t_score,df)

    #calculating Spooled for the calculation of the Cohens D value and later Hedges G value
    SDpooled = np.sqrt((s1**2*(n_cs-1) + s2**2*(n_ss-1))/(n_cs+n_ss-2))
    #using Cohens D value formula, we calculate that:
    Cohensd = (mean_cs - mean_ss)/SDpooled
    #calculating HedgesG as our sample population size is smaller than 30. We calculate it using Cohensd multipled by
    #(1-3/(4*(n1+n2)-9)
    HedgesG = Cohensd*(1-3/(4*(n_cs+n_ss)-9))

    #p-value
    print('p =',p_value)
    #effect size for both d and g
    print('d =',Cohensd)
    print('g = ', HedgesG)
dmt(test, control, tails=1) #setting talis to 1 as we are doing one side test becuase of our alternative hypothesis
```

```
p = 0.005406447039175174
d = 1.3580947476442466
g =  1.300710462532518
```

## References

VanderPlas, J. (2019). Combining Datasets: Merge and Join | Python Data Science Handbook. Retrieved 14 December 2019, from https://jakevdp.github.io/PythonDataScienceHandbook/03.07-merge-and-join.html

WSJ.com. (2017). *Salary Increase By Major,* 2017 [data file]. Retrieved 15 December 2019, from https://www.kaggle.com/wsj/college-salaries#degrees-that-pay-back.csv