Homework 01: Data Cleaning and Exploratory Data Analysis

Name: Julia Troni

This assignment is due on Canvas by **6:00PM on Friday September 2**. Your solutions to theoretical questions should be done in Markdown directly below the associated question. Your solutions to computational questions should include any specified Python code and results as well as written commentary on your conclusions. Remember that you are encouraged to discuss the problems with your classmates, but **you must write all code and solutions on your own**.

NOTES:

- Any relevant data sets should be available in the Homework 01 assignment write-up on Canvas. To make life easier on the grader if they need to run your code, do not change the relative path names here. Instead, move the files around on your computer.
- If you're not familiar with typesetting math directly into Markdown then by all means, do your work on paper first and then typeset it later. Remember that there is a reference guide linked on Canvas on writing math in Markdown. **All** of your written commentary, justifications and mathematical work should be in Markdown.
- Because you can technically evaluate notebook cells is a non-linear order, it's a good idea to do Kernel → Restart & Run All as a check before submitting your solutions. That way if we need to run your code you will know that it will work as expected.
- It is **bad form** to make your reader interpret numerical output from your code. If a question asks you to compute some value from the data you should show your code output **AND** write a summary of the results in Markdown directly below your code.
- This probably goes without saying, but... For any question that asks you to calculate something, you **must show all work and justify your answers to receive credit**. Sparse or nonexistent work will receive sparse or nonexistent credit.

Per usual you should import Pandas and NumPy when you start working with data.

In [22]:
```
import pandas as pd
import numpy as np
```

Also, lets read in the csv file. The path you use should reflect where you are keeping the data. For simplicity sake, put the data file (i.e. the .csv file) in the same folder as the Jupyter notebook file.

You will create a dataframe (df) called **CollegeType**. The standard is to precede the name with `df`, therefore call it `dfCollegeType` and read it in.

One way to do this (not recommended) is required when you store the data in a different location than the Jupyter file. See below.

The code is commented out; don't uncomment the cell and run it unless you are actually keeping your data in an alternate location than your Jupyter file for HW1.

In [23]:
```
# dfCollegeType = pd.read_csv("/location1/location2/location3/.../salaries-by-college-t
```

Now, if you are actually keeping the data in the same folder as the Jupyter file, we can read in the csv file with the following:

In [24]:
```
dfCollegeType = pd.read_csv("salaries-by-college-type.csv")
```

Problem 1

Lets look at the data you just read in to see how it appears, and decide how it needs cleaned (all data needs cleaned).

FYI, in the future always look over your data and clean it if it hasn't already been cleaned.

In [25]:
```
dfCollegeType
```

Out[25]:

	School Name	School Type	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
0	Massachusetts Institute of Technology (MIT)	Engineering	$72,200.00	$126,000.00	$76,800.00	$99,200.00	$168,000.00	$220,000.00
1	California Institute of Technology (CIT)	Engineering	$75,500.00	$123,000.00	NaN	$104,000.00	$161,000.00	NaN
2	Harvey Mudd College	Engineering	$71,800.00	$122,000.00	NaN	$96,000.00	$180,000.00	NaN
3	Polytechnic University of New York, Brooklyn	Engineering	$62,400.00	$114,000.00	$66,800.00	$94,300.00	$143,000.00	$190,000.00
4	Cooper Union	Engineering	$62,200.00	$114,000.00	NaN	$80,200.00	$142,000.00	NaN
...	...	...	...	...	...	...	...	...
264	Austin Peay State University	State	$37,700.00	$59,200.00	$32,200.00	$40,500.00	$73,900.00	$96,200.00

	School Name	School Type	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
265	Pittsburg State University	State	$40,400.00	$58,200.00	$25,600.00	$46,000.00	$84,600.00	$117,000.00
266	Southern Utah University	State	$41,900.00	$56,500.00	$30,700.00	$39,700.00	$78,400.00	$116,000.00
267	Montana State University - Billings	State	$37,900.00	$50,600.00	$22,600.00	$31,800.00	$78,500.00	$98,900.00
268	Black Hills State University	State	$35,300.00	$43,900.00	$27,000.00	$32,200.00	$60,900.00	$87,600.00

269 rows × 8 columns

Description of this data set

This data is a collection of median salaries from 269 institutions of various types.

The median salary was calculated at various times (Starting salary and mid-career) in the careers of graduates from these same institutions . For a particular year, each school reported the starting and mid-career salary for approximately 100 students. Each group of 100 was chosen with a random number generator from a list of alumni for that particular school.

Therefore, each school can list the median **starting salary** as well as the median **mid-career salary**. Also, along with the median (50th percentile) some schools also reported the 10th, the 25th, the 75th and the 90th percentile mid-career salaries .

Notice that some schools did not report all percentiles, hence the NAN's seen in the data.

Part A

Answer the following questions:

1] ***(2 points)*** What is the VOI (Variable of Interest) in this data set?

2] ***(2 points)*** What is the sample frame for this data set?

3] ***(2 points)*** What type of sample is this (convenience sample,systematic sample, census sample, stratified sample,...) ?

4] ***(2 points)*** What is the population and what is the sample?

My Answer:

1] Variable of Interest (VOI) is what we are trying to measure, so in this data set it is the median salary.

2] The sample frame is source from which the sample is drawn which is a list of 269 institutions containing alumni salaries.

3] This is a simple random sample because 100 students are randomly selected from each school. These 100 are randomly chosen with a random number generator.

4] The population is student alumni from various types of institutions.
The sample is subset of the population, and in this case the sample is group of 100 student alumni from each institution

Example help for cleaning (Required in Part B)

Now that we have an idea of what this data is and where it came from, we might want to construct some questions of interest like, "What type of school has the highest starting salary?"

However, before we can work with the data, it needs to be cleaned**!**

For starters, notice the numerical columns are strings with non-numerical symbols such as: `$`, `,`, `.`, and `NaN`

In **general**, here is a way to clean data similar to this set. All the string surgery is seen in the next cell:

In [26]:
```
#df[col] = df[col].replace('$', '')  # assuming there exist '$' symbols in the data
#df[col] = df[col].replace(',', '')  # assuming ',' is the thousand's separator in the
#df[col] = df[col].replace('%', '')  # assuming there exist '%' symbols in the data
## etc. for any other undesired symbols in the data
```

Then we would cast the data in the column as a float, as seen in the next cell:

In [27]:
```
#df[col] = df[col].astype(float)  # cast back to appropriate type
```

Specifically, you should run the following code for the column entitled 'Starting Median Salary' in this data set:

In [28]:
```
# Eliminate the ',' from the strings in the 'Starting Median Salary' column
dfCollegeType['Starting Median Salary'] = dfCollegeType['Starting Median Salary'].str.r

# Eliminate the '$' from the strings in the 'Startin Median Salary' column
dfCollegeType['Starting Median Salary'] = dfCollegeType['Starting Median Salary'].str.r

# Cast the strings in the 'Starting Median Salary' as floats.
dfCollegeType['Starting Median Salary'] = dfCollegeType['Starting Median Salary'].astyp
```

```
# Now take a look at the dataframe. The strings in the column 'Starting Median Salary'
dfCollegeType
```

Out[28]:

	School Name	School Type	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
0	Massachusetts Institute of Technology (MIT)	Engineering	72200.0	$126,000.00	$76,800.00	$99,200.00	$168,000.00	$220,000.00
1	California Institute of Technology (CIT)	Engineering	75500.0	$123,000.00	NaN	$104,000.00	$161,000.00	NaN
2	Harvey Mudd College	Engineering	71800.0	$122,000.00	NaN	$96,000.00	$180,000.00	NaN
3	Polytechnic University of New York, Brooklyn	Engineering	62400.0	$114,000.00	$66,800.00	$94,300.00	$143,000.00	$190,000.00
4	Cooper Union	Engineering	62200.0	$114,000.00	NaN	$80,200.00	$142,000.00	NaN
...	...	...	...	...	...	...	...	...
264	Austin Peay State University	State	37700.0	$59,200.00	$32,200.00	$40,500.00	$73,900.00	$96,200.00
265	Pittsburg State University	State	40400.0	$58,200.00	$25,600.00	$46,000.00	$84,600.00	$117,000.00
266	Southern Utah University	State	41900.0	$56,500.00	$30,700.00	$39,700.00	$78,400.00	$116,000.00
267	Montana State University - Billings	State	37900.0	$50,600.00	$22,600.00	$31,800.00	$78,500.00	$98,900.00
268	Black Hills State University	State	35300.0	$43,900.00	$27,000.00	$32,200.00	$60,900.00	$87,600.00

269 rows × 8 columns

Part B

Find the mean of the column 'Starting Median Salary' by hand.

The data in the column 'Starting Median Salary' is now "clean" in that calculations can be performed on the floats held in that column.

Let find the mean of this column by performing the hand calculation $\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$

Sum the numbers in the column and store them in a variable called `sum_xs`

```
sum_xs = dfCollegeType['Starting Median Salary'].sum()
```

Find out how many items are in the data frame and store that number in a variable called `n`

```
n = len(dfCollegeType.index)
```

(2 points) Create a variable `mean_SMS_byHand` . Perform the required calculation for a mean and print it to the screen.

In [32]:
```
sum_xs = dfCollegeType['Starting Median Salary'].sum()
n = len(dfCollegeType.index) # or n= dfCollegeType['Starting Median Salary'].count()
mean_SMS_byHand= sum_xs/n

print(mean_SMS_byHand)
```

```
46068.40148698885
```

Example help for Parts C and D

Now, lets find that same mean again by simply using the packaged `.mean` function.

Create variable 'mean_SMS' to hold the mean of the column 'Starting Median Salary'

You may want to notice the arguments for the `.mean` function:

`axis = 0` implies the **column** of data as opposed to the **row** of data. `axis = 1` implies row. `skipna` means to skip any data with an 'NAN' in it.

`axis = 0` is not necessary in this situation and `skipna = True` is the default and also need not be put here.

In [33]:
```
# We now put the mean of the column 'Starting Median Salary' into the variable `mean_SM

# mean_SMS = dfCollegeType[['Starting Median Salary']].mean(axis = 0, skipna = True)
# or
mean_SMS = dfCollegeType[['Starting Median Salary']].mean()

# Then we print 'mean_SMS' to the screen
print(mean_SMS)
```

```
Starting Median Salary    46068.401487
dtype: float64
```

Let's clean the data for the 'Mid-Career 10th Percentile Salary' column.

In [34]:

```
dfCollegeType['Mid-Career 10th Percentile Salary'] = dfCollegeType['Mid-Career 10th Per
dfCollegeType['Mid-Career 10th Percentile Salary'] = dfCollegeType['Mid-Career 10th Per
dfCollegeType['Mid-Career 10th Percentile Salary'] = dfCollegeType['Mid-Career 10th Per

# Now take a look at the dataframe. The strings in the column 'Mid-Career 10th Percenti
dfCollegeType
```

Out[34]:

	School Name	School Type	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
0	Massachusetts Institute of Technology (MIT)	Engineering	72200.0	$126,000.00	76800.0	$99,200.00	$168,000.00	$220,000.00
1	California Institute of Technology (CIT)	Engineering	75500.0	$123,000.00	NaN	$104,000.00	$161,000.00	NaN
2	Harvey Mudd College	Engineering	71800.0	$122,000.00	NaN	$96,000.00	$180,000.00	NaN
3	Polytechnic University of New York, Brooklyn	Engineering	62400.0	$114,000.00	66800.0	$94,300.00	$143,000.00	$190,000.00
4	Cooper Union	Engineering	62200.0	$114,000.00	NaN	$80,200.00	$142,000.00	NaN
...	...	...	...	...	...	...	...	...
264	Austin Peay State University	State	37700.0	$59,200.00	32200.0	$40,500.00	$73,900.00	$96,200.00
265	Pittsburg State University	State	40400.0	$58,200.00	25600.0	$46,000.00	$84,600.00	$117,000.00
266	Southern Utah University	State	41900.0	$56,500.00	30700.0	$39,700.00	$78,400.00	$116,000.00
267	Montana State University - Billings	State	37900.0	$50,600.00	22600.0	$31,800.00	$78,500.00	$98,900.00
268	Black Hills State University	State	35300.0	$43,900.00	27000.0	$32,200.00	$60,900.00	$87,600.00

269 rows × 8 columns

And finally lets print the mean for `Mid-Career 10th Percentile Salary` column

In [35]:
```
mean_MC10 = dfCollegeType[['Mid-Career 10th Percentile Salary']].mean(axis = 0, skipna
print(mean_MC10)
```

```
Mid-Career 10th Percentile Salary    44250.649351
dtype: float64
```

Part C

Clean the data for the following three columns:

1] ***(3 points)*** Clean the data in column 'Mid-Career Median Salary'

2] ***(3 points)*** Clean the data in column 'Mid-Career 25th Percentile Salary'

3] ***(3 points)*** Clean the data in column 'Mid-Career 75th Percentile Salary'

In [36]:
```
dfCollegeType[ 'Mid-Career Median Salary'] = dfCollegeType[ 'Mid-Career Median Salary']
dfCollegeType[ 'Mid-Career Median Salary'] = dfCollegeType[ 'Mid-Career Median Salary']
dfCollegeType[ 'Mid-Career Median Salary'] = dfCollegeType[ 'Mid-Career Median Salary']


dfCollegeType[ 'Mid-Career 25th Percentile Salary'] = dfCollegeType[  'Mid-Career 25th
dfCollegeType[  'Mid-Career 25th Percentile Salary'] = dfCollegeType[  'Mid-Career 25th
dfCollegeType[ 'Mid-Career 25th Percentile Salary'] = dfCollegeType[  'Mid-Career 25th


dfCollegeType[ 'Mid-Career 75th Percentile Salary'] = dfCollegeType[  'Mid-Career 75th
dfCollegeType[  'Mid-Career 75th Percentile Salary'] = dfCollegeType[  'Mid-Career 75th
dfCollegeType[ 'Mid-Career 75th Percentile Salary'] = dfCollegeType[  'Mid-Career 75th


dfCollegeType[ 'Mid-Career 90th Percentile Salary'] = dfCollegeType[  'Mid-Career 90th
dfCollegeType[  'Mid-Career 90th Percentile Salary'] = dfCollegeType[  'Mid-Career 90th
dfCollegeType[ 'Mid-Career 90th Percentile Salary'] = dfCollegeType[  'Mid-Career 90th


# Now take a look at the dataframe. All salary columns should be floats.
dfCollegeType
```

Out[36]:

	School Name	School Type	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
0	Massachusetts Institute of Technology (MIT)	Engineering	72200.0	126000.0	76800.0	99200.0	168000.0	220000.0

	School Name	School Type	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
1	California Institute of Technology (CIT)	Engineering	75500.0	123000.0	NaN	104000.0	161000.0	NaN
2	Harvey Mudd College	Engineering	71800.0	122000.0	NaN	96000.0	180000.0	NaN
3	Polytechnic University of New York, Brooklyn	Engineering	62400.0	114000.0	66800.0	94300.0	143000.0	190000.0
4	Cooper Union	Engineering	62200.0	114000.0	NaN	80200.0	142000.0	NaN
...	...	...	...	...	...	...	...	...
264	Austin Peay State University	State	37700.0	59200.0	32200.0	40500.0	73900.0	96200.0
265	Pittsburg State University	State	40400.0	58200.0	25600.0	46000.0	84600.0	117000.0
266	Southern Utah University	State	41900.0	56500.0	30700.0	39700.0	78400.0	116000.0
267	Montana State University - Billings	State	37900.0	50600.0	22600.0	31800.0	78500.0	98900.0
268	Black Hills State University	State	35300.0	43900.0	27000.0	32200.0	60900.0	87600.0

269 rows × 8 columns

Part D

Find the means for the same columns cleaned in part C.

Variable names should be:

1] ***(2 points)*** `mean_MCM` for the mean of 'Mid-Career Median Salary' column values.

2] ***(2 points)*** `mean_MC25` for the mean of 'Mid-Career 25th Percentile Salary' column values.

3] ***(2 points)*** `mean_MC75` for the mean of 'Mid-Career 75th Percentile Salary' column values.

In [37]:
```
mean_MCM = dfCollegeType[['Mid-Career Median Salary']].mean(axis = 0, skipna = True)
```

```
print(mean_MCM)

mean_MC25 = dfCollegeType[['Mid-Career 25th Percentile Salary']].mean(axis = 0, skipna
print(mean_MC25)


mean_MC75 = dfCollegeType[['Mid-Career 75th Percentile Salary']].mean(axis = 0, skipna
print(mean_MC75)

mean_MC90 = dfCollegeType[['Mid-Career 90th Percentile Salary']].mean(axis = 0, skipna
print(mean_MC90)
```

```
Mid-Career Median Salary    83932.342007
dtype: float64
Mid-Career 25th Percentile Salary    60373.234201
dtype: float64
Mid-Career 75th Percentile Salary    116275.092937
dtype: float64
Mid-Career 90th Percentile Salary    157705.627706
dtype: float64
```

Problem 2

So now we have a single number (mean) that describes the center of the data provided in each column. But what technically did we just find?

Answer: The mean of all schools' median salaries; either starting or mid-career. This may not be of the utmost interest to us though since all schools are lumped into this one number.

Let's instead seperate the types of institutions to find out what type of school has the highest starting salary, according to the medians provided.

But how do we know how many types of schools are in this dataframe without spending time fishing through the whole set?

Try one of the following two snippets of code:

`dfCollegeType["School Type"].unique()`

The `.unique` function will list the unique qualifiers in the column, or

`set(dfCollegeType["School Type"].values)`

Using both `set` and `values` will give you a set of the values listed in the column.

Part A

(1 point) Run one of those code snippets in the cell below:

In [38]:
```
dfCollegeType["School Type"].unique()
```

Out[38]:
```
array(['Engineering', 'Party', 'Liberal Arts', 'Ivy League', 'State'],
      dtype=object)
```

Part B

(1 point) **Question: How many school types are there? List them.**

- There are 5 types of schools:
    1. Engineering
    2. Party
    3. Liberal Arts
    4. Ivy League
    5. State

Example help for Part C

If you wanted the mean of the starting median salaries for 'Ivy League' schools then you would apply the `.mean` function to **only** those 'starting median salaries for Ivy League schools.

This can be done with `.loc` as follows:

In [39]:
```
# Create a variable called `ivy_mean`
ivy_mean = dfCollegeType.loc[dfCollegeType['School Type'] == 'Ivy League', 'Starting Me
print("The average (mean) of the starting median salaries for Ivy League schools is ",
```

```
The average (mean) of the starting median salaries for Ivy League schools is  60475.0
```

Part C

1] ***(2 points)*** Find the mean of the starting median salaries for **Engineering schools**.

2] ***(2 points)*** Find the mean of the starting median salaries for **State schools.**

3] ***(2 points)*** Find the mean of the starting median salaries for **Liberal Arts schools**.

(These means need not be edited to dollars and cents from floats)

In [40]:
```
eng_mean = dfCollegeType.loc[dfCollegeType['School Type'] == 'Engineering', 'Starting M
print("The average (mean) of the starting median salaries for Engineering schools is ",

state_mean = dfCollegeType.loc[dfCollegeType['School Type'] == 'State', 'Starting Media
print("The average (mean) of the starting median salaries for State schools is ", state

libArt_mean = dfCollegeType.loc[dfCollegeType['School Type'] == 'Liberal Arts', 'Starti
print("The average (mean) of the starting median salaries for Liberal Arts schools is "
```

```
The average (mean) of the starting median salaries for Engineering schools is  59057.894
73684211
The average (mean) of the starting median salaries for State schools is  44126.285714285
```

```
72
The average (mean) of the starting median salaries for Liberal Arts schools is  45746.80
85106383
```

Part D

(6 points) Question: About how much can a person expect to make in the 90th percentile if they are mid-career and got their degree from a party school?

i.e. report the mean of the mid-career 90th percentile salaries for party schools.

In [41]:
```
# Code your solution to part D here
party_mean = dfCollegeType.loc[dfCollegeType['School Type']=="Party", "Mid-Career 90th
print("Party school and mid-career, one can expect to make {} in the 90th percentile".f
```

```
Party school and mid-career, one can expect to make 166947.36842105264 in the 90th perce
ntile
```

Problem 3

Now we would like to use the `.describe` function on our dataframe.

(3 points) Type the following snippet of code into the coding cell below:
`dfCollegeType.describe()`

For full points you want to fully describe what you find in `.describe`. For instance, describe the output of the code and determine if the 5-number summary is included in the output. Why is `count` different for some of the columns? Is `mean` the same as the values calculated above? What is `std`? etc.

In [42]:
```
dfCollegeType.describe()
```

Out[42]:

	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
count	269.000000	269.000000	231.000000	269.000000	269.000000	231.000000
mean	46068.401487	83932.342007	44250.649351	60373.234201	116275.092937	157705.627706
std	6412.616242	14336.191107	8719.612427	11381.348857	22952.334054	34823.348157
min	34800.000000	43900.000000	22600.000000	31800.000000	60900.000000	87600.000000
25%	42000.000000	74000.000000	39000.000000	53200.000000	100000.000000	136000.000000
50%	44700.000000	81600.000000	43100.000000	58400.000000	113000.000000	153000.000000
75%	48300.000000	92200.000000	47400.000000	65100.000000	126000.000000	170500.000000
max	75500.000000	134000.000000	80000.000000	104000.000000	234000.000000	326000.000000

Analysis of what this data is showing us

- Since this is numeric data, the `.describe` method includes the count, mean, standard deviation, minimum, maximum as well as lower, 50 and upper percentiles of each column. By default the lower percentile is 25 and the upper percentile is 75. The 50 percentile is the same as the median.
- The `.describe` method is particularly useful for easily creating a box and whisker plot. It also is helpful for examining the skewedness of the data (particularly by comparing the median to the max and min)
- The `count` is different for some of the columns because the `.count` method only counts the number of non-NA/null observations. Since there are 38 NA entries in each the 10th and 90th percentile columns, the count is only 231 as opposed to 269. This means that 38 schools did NOT submit data for Mid-Career 10th Percentile Salary and Mid-Career 90th Percentile Salary.
- The `mean` calculated with `.describe` is the same as the values that I calculated above
- The `std` row is calculating the standard deviation of the data for each column. By default, pandas normalizes by N-1, meaning the standard deviation is calculated for the sample. This is what we want because this is not a census sample type, so we do not want to calculate std for a population.

Problem 4

Let $x_1, x_2, \ldots, x_n$ be n observations of a variable of interest. Recall that the sample mean $\bar{x}$ and sample variance s^2 are given by

$$\bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k \quad \text{and} \quad s^2 = \frac{1}{n-1}\sum_{k=1}^{n} (x_k - \bar{x})^2$$

Recall standard deviation is $\sqrt{s^2} = s$, i.e. the square root of the variance.

Computationally, $s^2 = \frac{1}{n-1}\sum_{k=1}^{n} (x_k - \bar{x})^2$ reveals an issue with data manipulation. This formula, when coded, would require two passes over the data: one to compute the mean, and a second to subtract the mean from each observation and compute the sum of squares.

Additionally, this manner of computing sample variance can lead to numerical problems and inefficiencies. For example, if the x's are "large" and the differences between them "small", computing the sample variance using the formula above requires computing a small number as the difference of two small numbers. Not good! Check out this optional wikipedia article about Loss of Significance to learn more.

Therefore, it is useful to be able to compute the variance and/or standard deviation in a single pass, inspecting each value x_k only once; for example, when the data are being collected without enough storage to keep all the values, or when costs of memory access dominate those of computation.

We need to manipulate the expressions algebraically in order to arrive at a more computationally efficient formula.

One such identity has been found; it is seen here:

$$s^2 = \frac{1}{n(n-1)}\left(n \cdot \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right)$$

(8 points) Prove that the two formulas for standard deviation will in fact calculate the same thing, i.e. prove they are the same formula.

Prove

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

and

$$s^2 = \frac{1}{n(n-1)}\left(n \cdot \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right)$$

are the same thing.

Solution:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

$$s^2 = \frac{1}{(n-1)}\sum_{i=1}^{n}(x_i^2 - 2x_i\bar{x} + \bar{x}^2)$$

$$s^2 = \frac{1}{(n-1)}\sum_{i=1}^{n} x_i^2 - \frac{1}{(n-1)}\sum_{i=1}^{n} 2x_i\bar{x} + \frac{1}{(n-1)}\sum_{i=1}^{n}\bar{x}^2$$

$$s^2 = \frac{1}{(n-1)}\sum_{i=1}^{n} x_i^2 - \frac{2\bar{x}}{(n-1)}\sum_{i=1}^{n} x_i + \frac{1}{(n-1)}\bar{x}^2$$

$$s^2 = \frac{1}{(n-1)}\sum_{i=1}^{n} x_i^2 - \frac{2(\frac{1}{n}\sum_{i=1}^{n} x_i)}{(n-1)}\sum_{i=1}^{n} x_i + \frac{1}{(n-1)}\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2$$

$$s^2 = \frac{1}{(n-1)}\sum_{i=1}^{n} x_i^2 - \frac{2}{n(n-1)}\left(\sum_{i=1}^{n} x_i\right)^2 + \frac{1}{n(n-1)}\left(\sum_{i=1}^{n} x_i\right)^2$$

$$s^2 = \frac{1}{(n-1)}\sum_{i=1}^{n} x_i^2 - \frac{1}{n(n-1)}\left(\sum_{i=1}^{n} x_i\right)^2$$

Thus:

$$s^2 = \frac{1}{n(n-1)} \left(n \cdot \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2 \right)$$