

**Northeastern University**

**EECE5642: Data Visualization  
Homework 2**

**Rudra Patel  
Feb 21, 2022**

## Answer 1

In this question we need to find a visualization work and use the principles and theories we learnt in class to analyze the work. Further I have suggested a step-by-step way to improve this depiction while giving insights and explanation for each new change.

The visualization work that I have used can be found at the listed source:  
<https://www.flickr.com/photos/gdsdigital/4963409391>

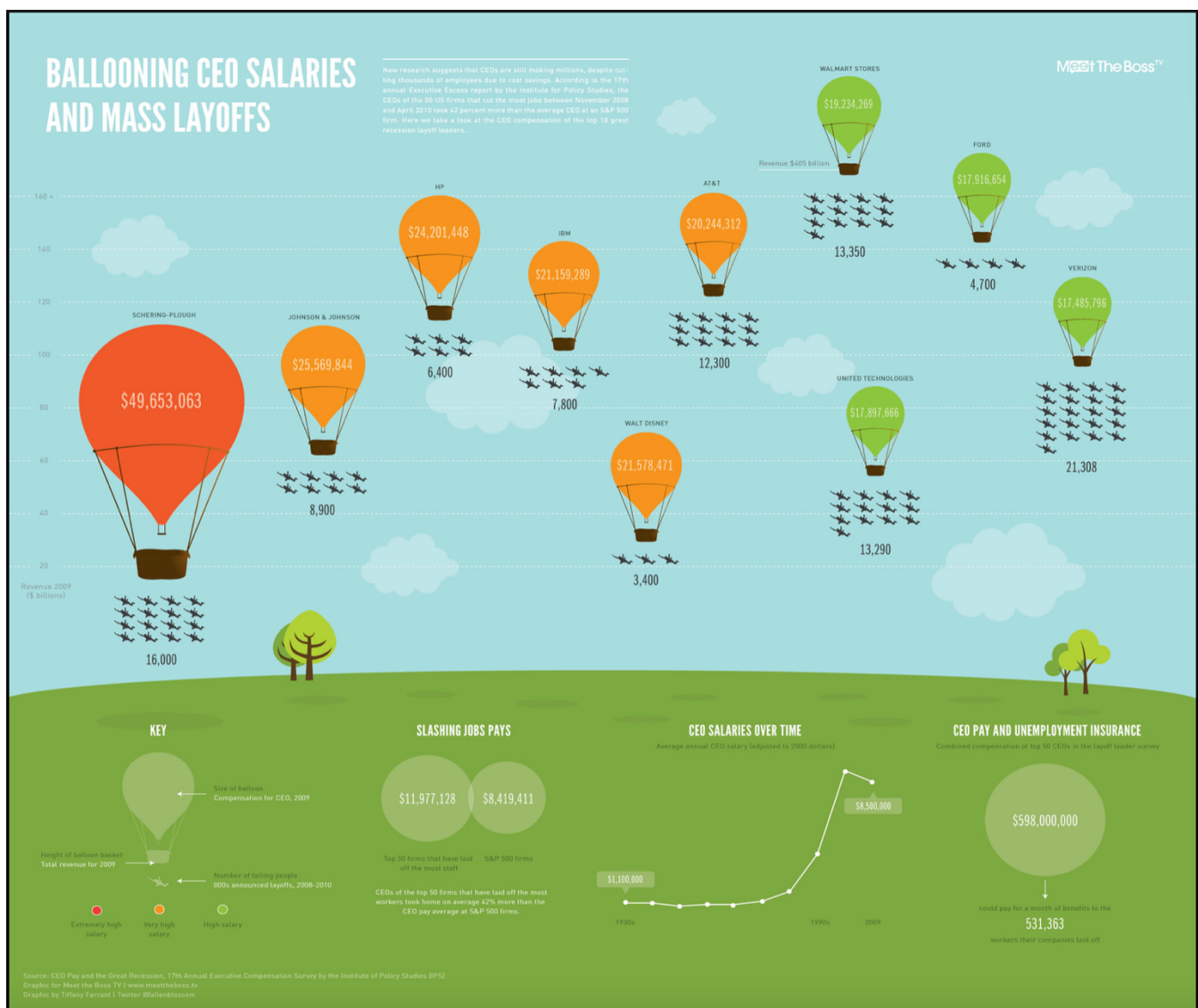


Figure 1. Original Visualization work

### ***Step-by-step analysis***

As a layman one would feel this is a good depiction with use of different graphics and multiple colors, but as a data analyst this might seem too fancy. Let us see what are the factors that lead to the consideration of this chart as a weak visualization work.

#### **1. Lie-factor:**

The lie factor is a ratio used to describe the relationship between the size of an effect in a graph and a size of an effect in data. The proportions of the quantities depicted should be directly proportional to the representation of numbers as measured on the graphic's surface.

$$\text{Lie factor} = \frac{\text{Size of effect shown in graphic}}{\text{Size of effect in data}}$$

$$\text{size of effect} = \frac{|\text{second value} - \text{first value}|}{\text{first value}}$$

In a good visualization work the value of **lie factor should be between 0.95 and 1.05** to ensure the integrity in the graphic. If the value is less or greater than this range it tells us that there are substantial distortions in the visualization.

In the original work it was found that there is a **bigger lie factor** which is the first factor leading to a negative rating of the visualization work.

In this chart the radius of the balloon is used to encode the graphic in the data.

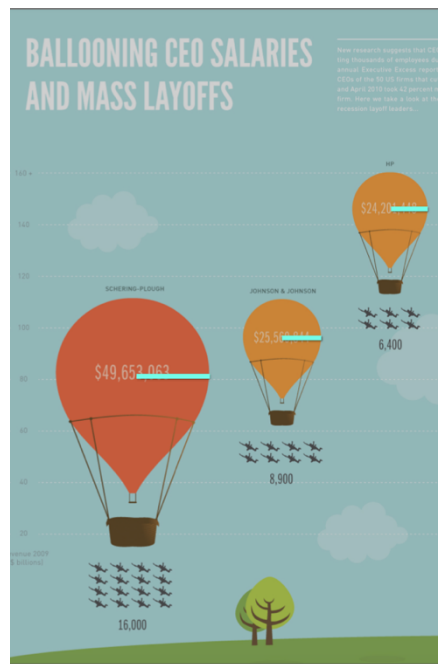


Figure 2. Depiction for Lie factor calculation

As shown in the above picture we measure the radius of the balloon and take a ratio of the change with respect to the quantities it depicts.

Since the **area of a circle as we know is  $\pi r^2$**  we can see that the change of radius from one balloon to the other balloon results in doubling of the area for every one unit increase of radius. This results in a **lie factor ratio 3** which is considerably **higher** and out of the desirable range of **0.95 - 1.05**.

The **true size of the balloon** should be as shown in the below chart.

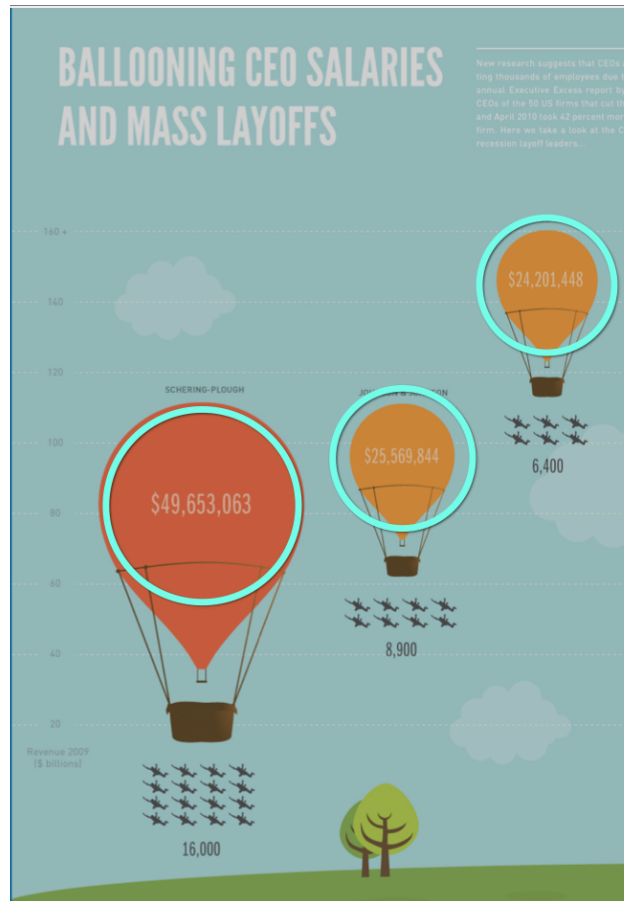


Figure 3. Depiction of modified balloon size

The above figure reflects how a change in the compensation for the CEO of a company in the year 2009 should reflect the change in size of the balloon shown. This modification will lead to a **small lie factor** which is now in the desirable range and would be a good visualization principle on its first place.

## 2. Data-Ink ratio:

As stated by Tufte the data ink ratio reflects the amount of data ink that is used to show some actual data as compared to ink that is used up for the entire visualization. A good work should only have **reasonable use of ink** which depicts something on the chart that is **directly related** to quantities or size to be drawn out for explanation.

This is extremely important as we need to **avoid distracting** the viewers towards unnecessary graphics which will have no meaning to the data we want to visualize. The end goal is to **maximize the data-ink ratio and make it as close as possible to 1** without compromising on the entities that are useful for effective communication.

As shown in the original image there several graphics used in this work that could have been avoided are not necessary to convey some information.

For example, the trees shown in the picture, the clouds shown in the sky and the ink itself used to show the grass or the sky are all not relevant. Even the balloon shown for that matter can be avoided by simply using histograms or lines to show the numbers.

Also, the number of people kicked off from the balloon have no real scale value, like for the 1<sup>st</sup> balloon the total number of layoffs are 16000 and 8 people are shown jumping out of the balloon which tells us roughly 1 graphic people equals 4000 number of real people where in the 2<sup>nd</sup>, 3<sup>rd</sup> or any subsequent balloon that same scale is not followed. All these combined makes for the second factor of a weak visualization work.

### **3. Tufte's Principle:**

The above 2 factors along with Tufte's principle all lean towards labelling the original graphic as a weak visualization. There are many points laid down by Tufte in order to create a good visualization. All these points are taken care of while designing a new suggested plot.

- Show the data
- Induce the viewer to think about the substance of the findings rather than the methodology, the graphical design, or other aspects
- Avoid distorting what the data have to say
- Present many numbers in a small space, i.e., efficiently
- Make large data sets coherent
- Encourage the eye to compare different pieces of data
- Reveal the data at several levels of detail, from a broad overview to the fine structure
- Serve a clear purpose: description, exploration, tabulation, or decoration
- Be closely integrated with the statistical and verbal descriptions of the data set

From E. R. Tufte. The Visual Display of Quantitative Information, 2nd Edition. Graphics Press, Cheshire, Connecticut, 2001.

Figure 4. Design Principles by Tufte

In the original graph there is a faint line showing the revenue of the company in 2009 which is irrelevant to the aim of the graph which is titled as “Ballooning CEO salaries and Mass Layoffs”. This can be removed from the graph along with the previously spotted unwanted graphics.

Next, we will show **various outputs step-by-step of the new suggested design** and give out valid reason to choose each of them respectively.

### **Multiple Bar Plot**

A multiple bar plot can be used to plot given data with the different companies listed on the x-axis and the CEO compensation and the total employee layoffs on the y-axis. A twin axis plot is a smart choice here since the two y-axis data scales have highly varying magnitude. Below shown is a multiple bar plot against the original visualization work.

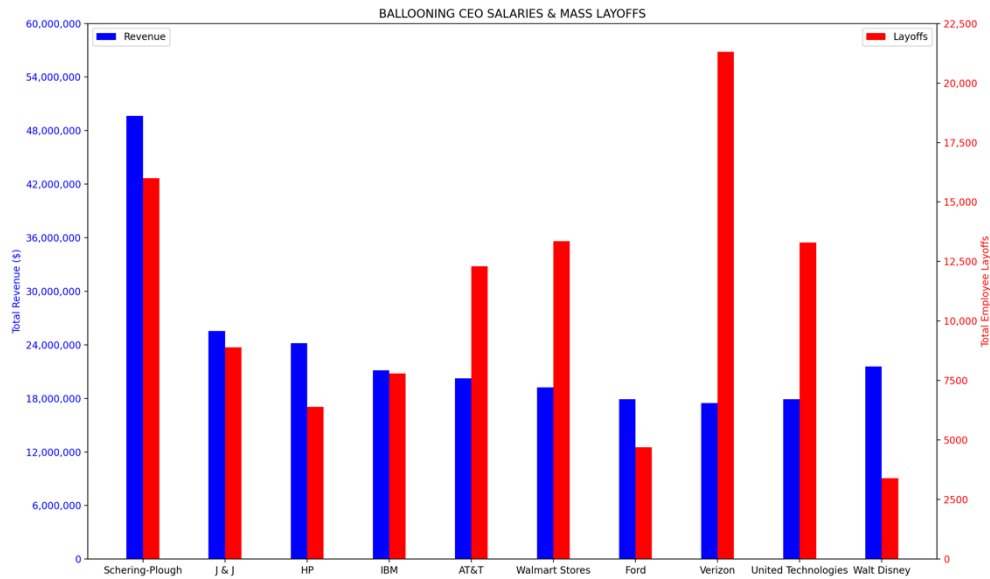


Figure 5. Multiple Bar plot

### Line plot

A line plot is useful to show the changing trend with different companies. The same way as in multiple bars plot the different companies are shown on x-axis and twin axis is used to show the two y-axis data values as shown below.

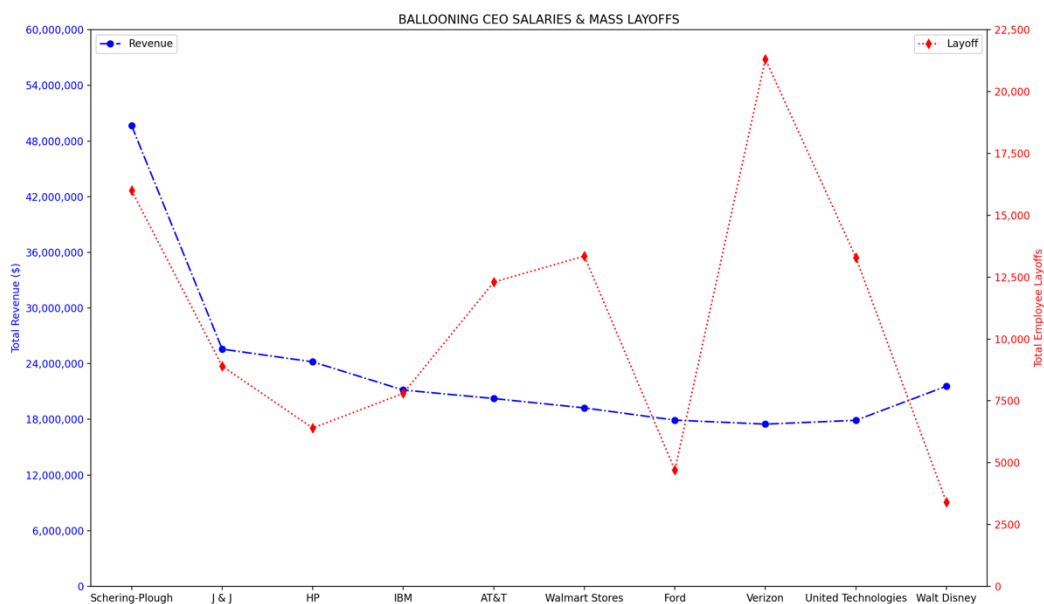


Figure 6. Line plot



Sometimes using multiple y-axis (twin axis) on a same plot can be difficult to interpret which results in using subplots which makes a depiction more standard and clearer as shown in the below figure.

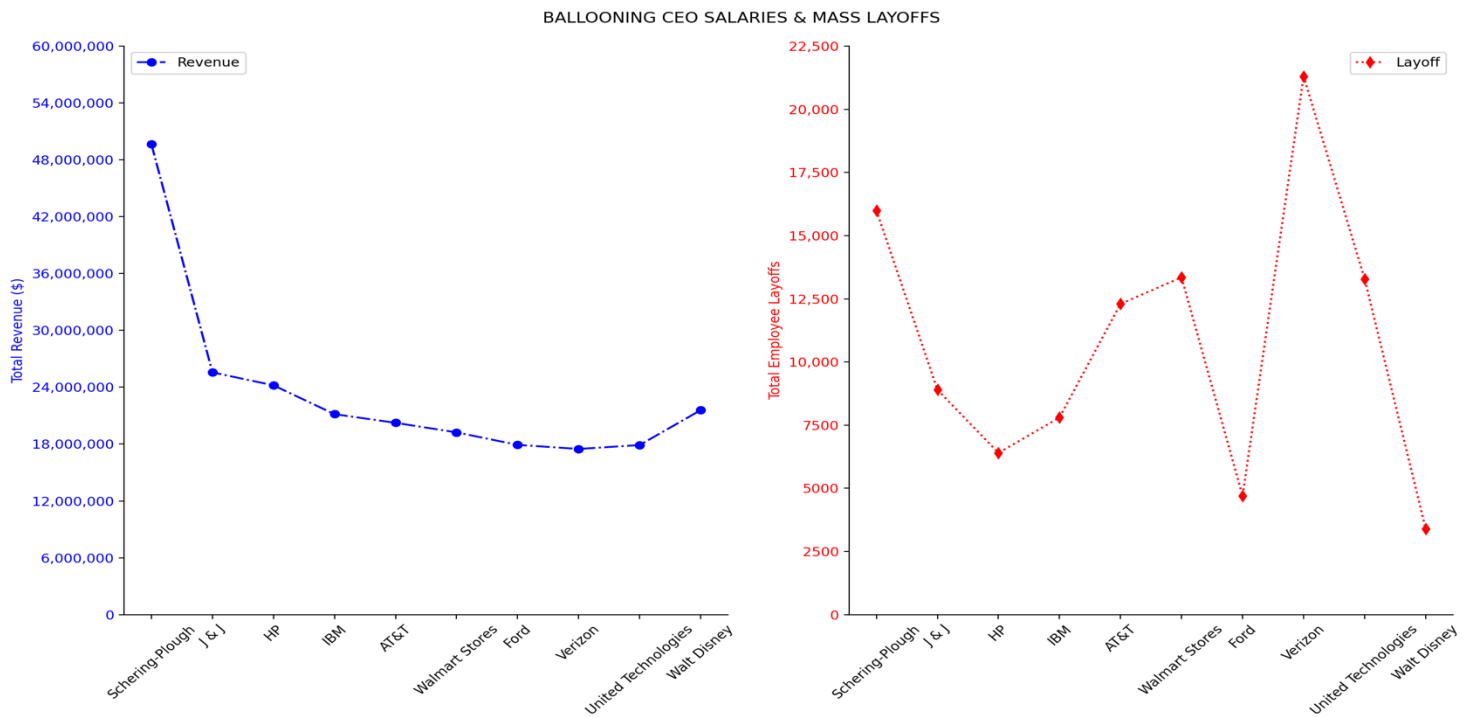


Figure 7. Line plot with separate subplot

This helps us in having a clearer understanding of both the y-axis since they are now plotted separately. For example, for the company **IBM** since the data points shown in Fig 6. are very **close** which might lead to a **confusion when interpreting the y-axis**.

The disadvantage in this type of chart is that it takes up **more space** as compared to the one where we plot both on the same graph.

## Scatter Plot

A more compact scatter plot can be used to **eliminate the disadvantage** seen above about space complexity. In this type of chart, we depict each company as a data point with the total number of employee layoffs on the x-axis and the compensation for CEO on the y-axis.

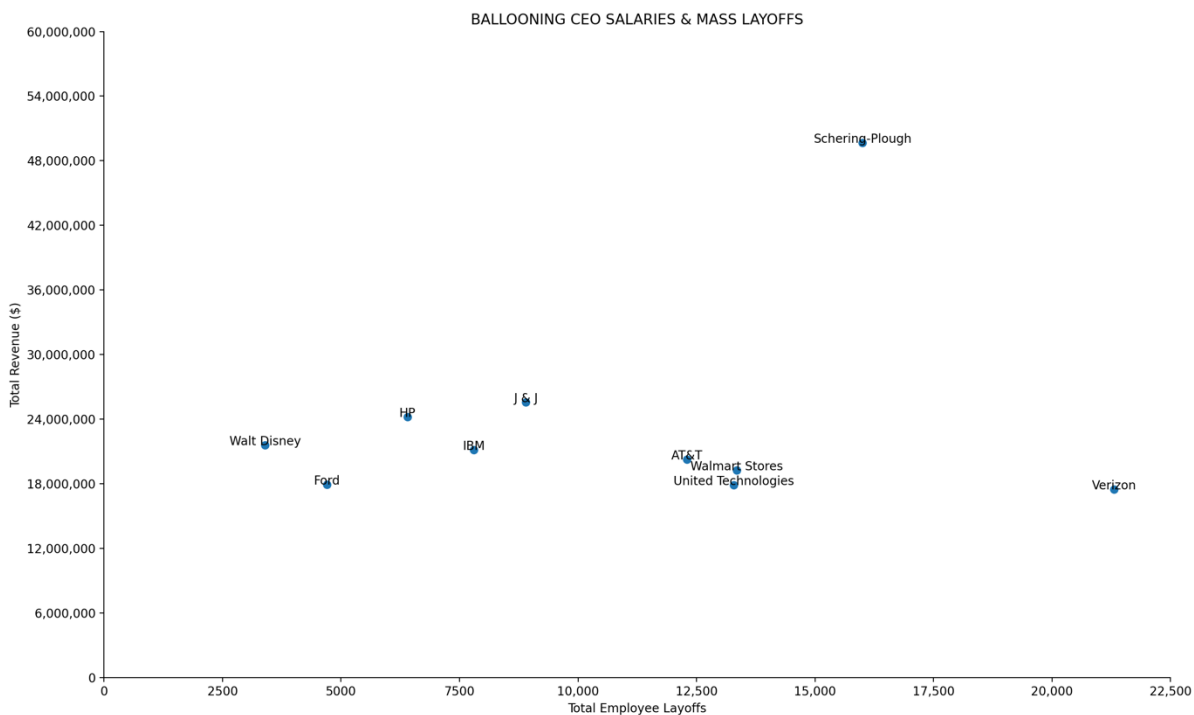


Figure 8. Scatter plot

**This is the final improved plot which focuses on plotting both the data with equal importance and clear vision. This also takes up less data-ink and overall, less space for visualization.**

*In all the above plots shown above an attempt has been made to integrate all the principles along with proper Lie factor ratio and maximum Data-ink in order to generate a clearer, better and more useful visualization work.*

## Answer 2

Input Image value:

$[R, G, B] = [137, 56, 146]$ ----- (in 0 – 255 range)

$[R, G, B] = [0.53725, 0.21961, 0.57255]$ ----- (in 0 – 1 range)



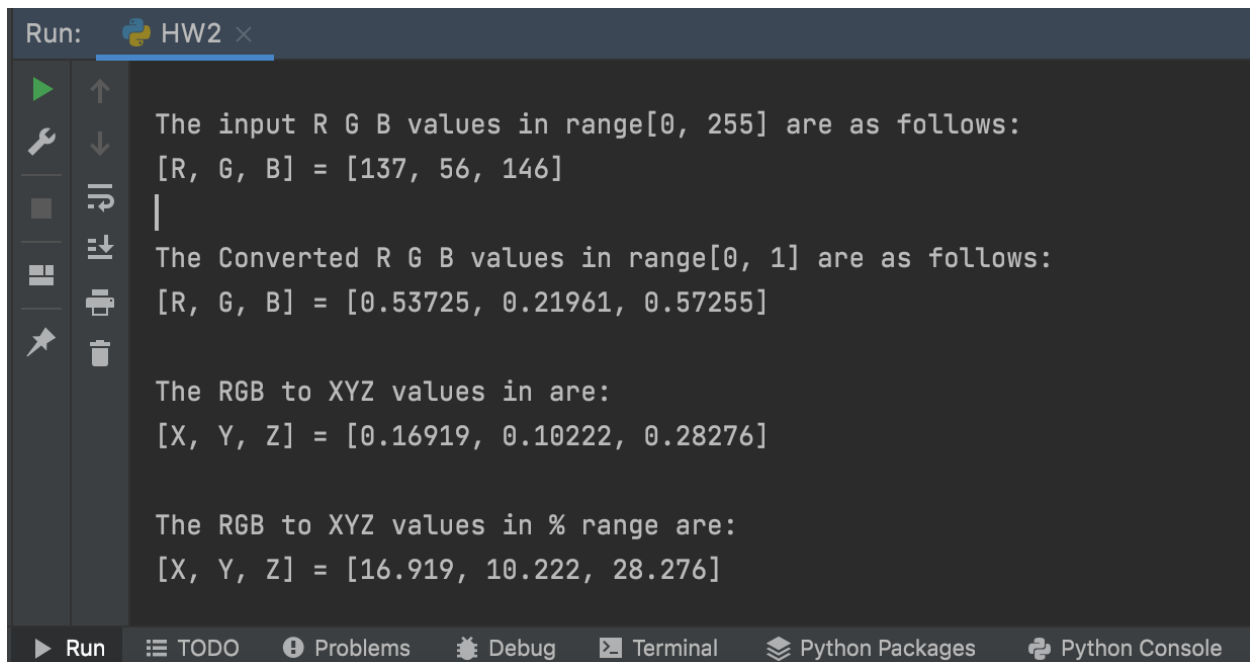
Figure 9. Input image

Now, below we will see **each conversion**. Each of the outputs have been generated by writing a python script and the python output is shown directly for each case.

The reference to this can be found at the source listed here:

<http://www.easyrgb.com/index.php?X=MATH>

## 1. RGB to CIE-XYZ



```
Run: HW2 x
The input R G B values in range[0, 255] are as follows:
[R, G, B] = [137, 56, 146]
|
The Converted R G B values in range[0, 1] are as follows:
[R, G, B] = [0.53725, 0.21961, 0.57255]

The RGB to XYZ values in are:
[X, Y, Z] = [0.16919, 0.10222, 0.28276]

The RGB to XYZ values in % range are:
[X, Y, Z] = [16.919, 10.222, 28.276]
```

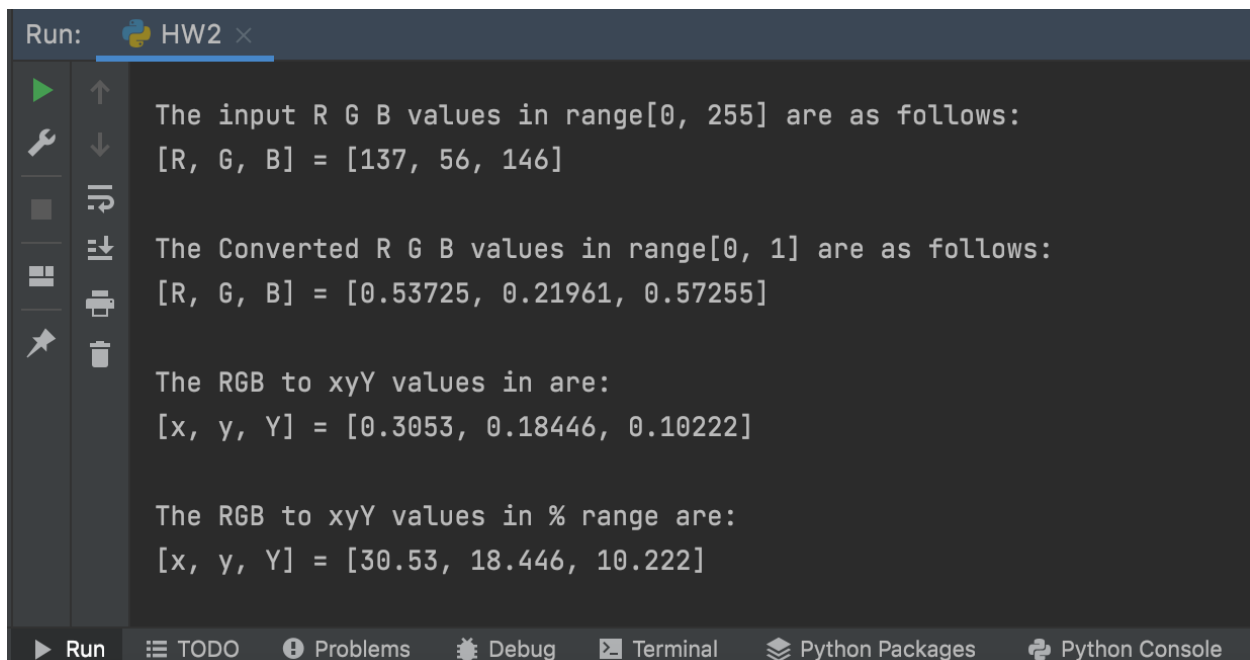
The screenshot shows a Python IDE terminal window titled 'Run: HW2 x'. The output text is as follows:

The input R G B values in range[0, 255] are as follows:  
[R, G, B] = [137, 56, 146]  
|  
The Converted R G B values in range[0, 1] are as follows:  
[R, G, B] = [0.53725, 0.21961, 0.57255]  
  
The RGB to XYZ values in are:  
[X, Y, Z] = [0.16919, 0.10222, 0.28276]  
  
The RGB to XYZ values in % range are:  
[X, Y, Z] = [16.919, 10.222, 28.276]

The IDE interface includes a sidebar with icons for Run, TODO, Problems, Debug, Terminal, Python Packages, and Python Console. The bottom status bar shows 'Run', 'TODO', 'Problems', 'Debug', 'Terminal', 'Python Packages', and 'Python Console'.

Figure 10. RGB to CIE-XYZ conversion output

## 2. RGB to CIE-xyY



```
Run: HW2 x
The input R G B values in range[0, 255] are as follows:
[R, G, B] = [137, 56, 146]

The Converted R G B values in range[0, 1] are as follows:
[R, G, B] = [0.53725, 0.21961, 0.57255]

The RGB to xyY values in are:
[x, y, Y] = [0.3053, 0.18446, 0.10222]

The RGB to xyY values in % range are:
[x, y, Y] = [30.53, 18.446, 10.222]
```

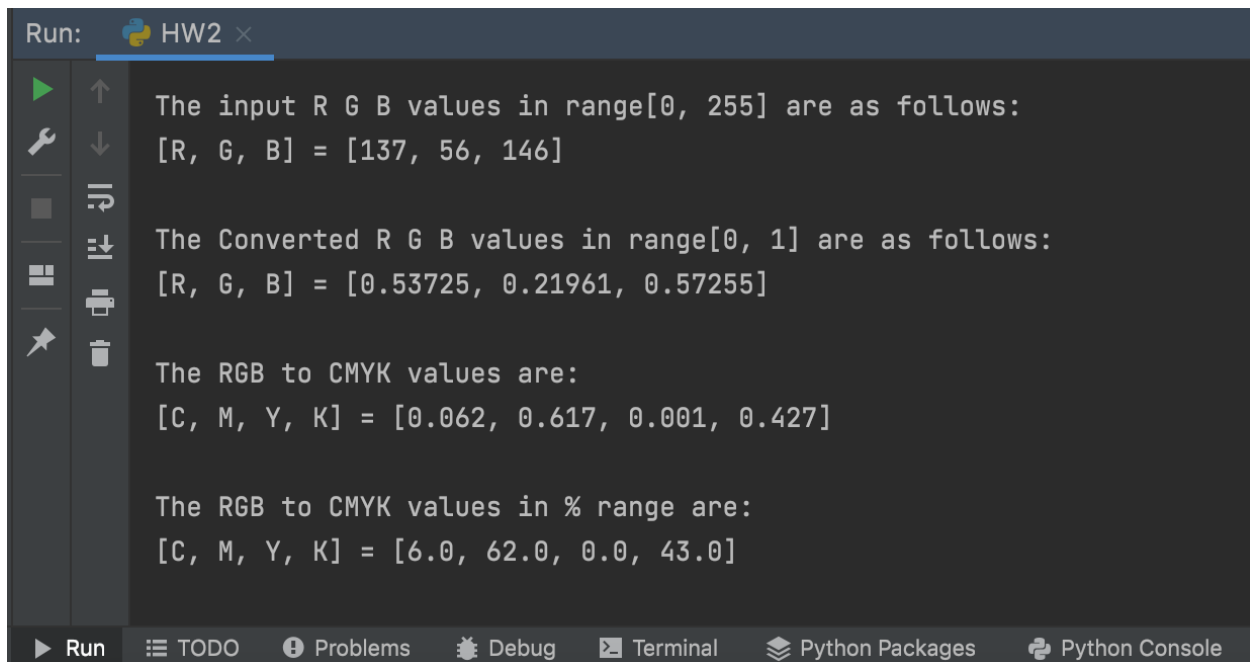
The screenshot shows a Python IDE terminal window titled 'Run: HW2 x'. The output text is as follows:

The input R G B values in range[0, 255] are as follows:  
[R, G, B] = [137, 56, 146]  
  
The Converted R G B values in range[0, 1] are as follows:  
[R, G, B] = [0.53725, 0.21961, 0.57255]  
  
The RGB to xyY values in are:  
[x, y, Y] = [0.3053, 0.18446, 0.10222]  
  
The RGB to xyY values in % range are:  
[x, y, Y] = [30.53, 18.446, 10.222]

The IDE interface includes a sidebar with icons for Run, TODO, Problems, Debug, Terminal, Python Packages, and Python Console. The bottom status bar shows 'Run', 'TODO', 'Problems', 'Debug', 'Terminal', 'Python Packages', and 'Python Console'.

Figure 11. RGB to CIE-xyY conversion output

### 3. RGB to CMYK



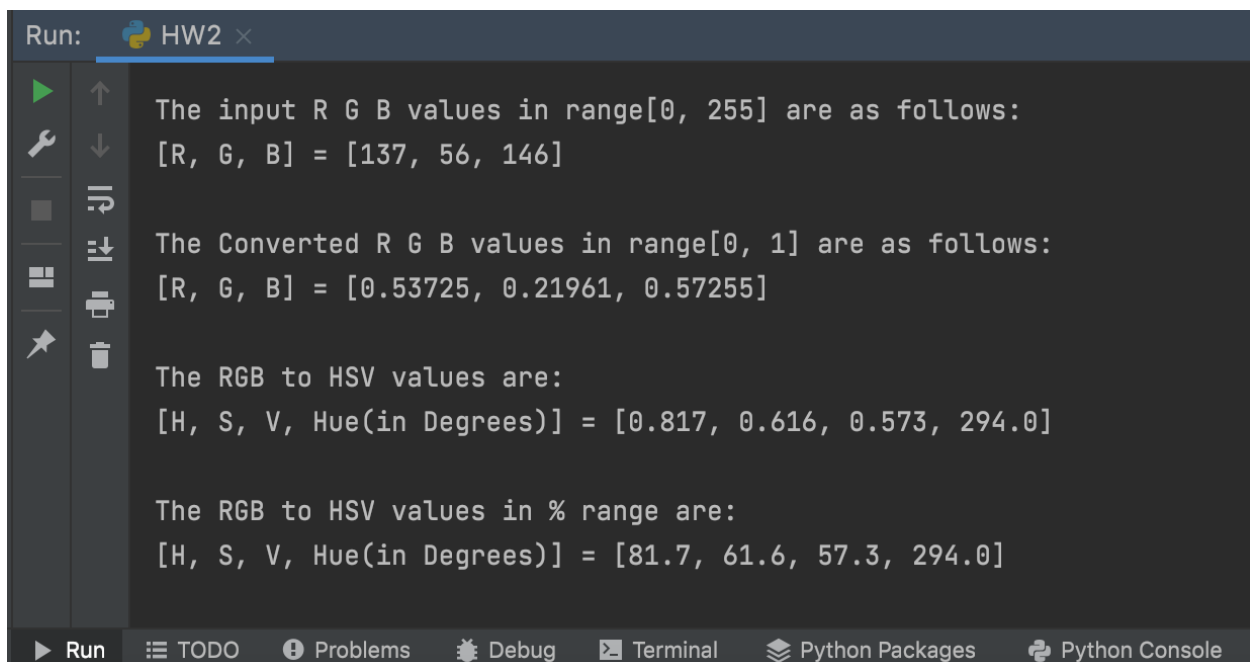
The screenshot shows a Python IDE console window titled 'Run: HW2'. The console output displays the following text:

```
The input R G B values in range[0, 255] are as follows:  
[R, G, B] = [137, 56, 146]  
  
The Converted R G B values in range[0, 1] are as follows:  
[R, G, B] = [0.53725, 0.21961, 0.57255]  
  
The RGB to CMYK values are:  
[C, M, Y, K] = [0.062, 0.617, 0.001, 0.427]  
  
The RGB to CMYK values in % range are:  
[C, M, Y, K] = [6.0, 62.0, 0.0, 43.0]
```

The IDE interface includes a sidebar with icons for Run, TODO, Problems, Debug, Terminal, Python Packages, and Python Console. The bottom status bar shows 'Run', 'TODO', 'Problems', 'Debug', 'Terminal', 'Python Packages', and 'Python Console'.

Figure 12. RGB to CMYK conversion output

### 4. RGB to HSV



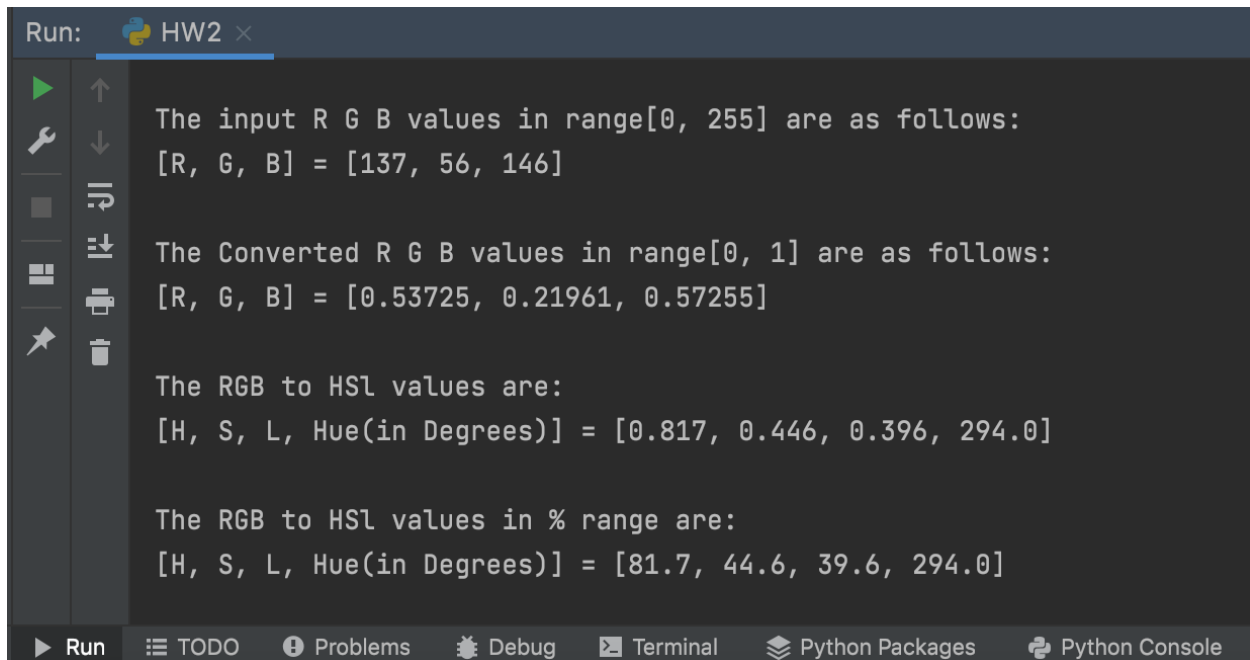
The screenshot shows a Python IDE console window titled 'Run: HW2'. The console output displays the following text:

```
The input R G B values in range[0, 255] are as follows:  
[R, G, B] = [137, 56, 146]  
  
The Converted R G B values in range[0, 1] are as follows:  
[R, G, B] = [0.53725, 0.21961, 0.57255]  
  
The RGB to HSV values are:  
[H, S, V, Hue(in Degrees)] = [0.817, 0.616, 0.573, 294.0]  
  
The RGB to HSV values in % range are:  
[H, S, V, Hue(in Degrees)] = [81.7, 61.6, 57.3, 294.0]
```

The IDE interface includes a sidebar with icons for Run, TODO, Problems, Debug, Terminal, Python Packages, and Python Console. The bottom status bar shows 'Run', 'TODO', 'Problems', 'Debug', 'Terminal', 'Python Packages', and 'Python Console'.

Figure 13. RGB to HSV conversion output

## 5. RGB to HSL



The screenshot shows a terminal window titled 'Run: HW2 x' with a dark background. On the left is a vertical toolbar with icons for running, debugging, and other IDE functions. The terminal output is as follows:

```
The input R G B values in range[0, 255] are as follows:  
[R, G, B] = [137, 56, 146]  
  
The Converted R G B values in range[0, 1] are as follows:  
[R, G, B] = [0.53725, 0.21961, 0.57255]  
  
The RGB to HSL values are:  
[H, S, L, Hue(in Degrees)] = [0.817, 0.446, 0.396, 294.0]  
  
The RGB to HSL values in % range are:  
[H, S, L, Hue(in Degrees)] = [81.7, 44.6, 39.6, 294.0]
```

At the bottom of the terminal window is a horizontal toolbar with buttons for 'Run', 'TODO', 'Problems', 'Debug', 'Terminal', 'Python Packages', and 'Python Console'.

Figure 14. RGB to HSL conversion output

***The various conversion codes and calculations can be found in the python script uploaded along with other code in a zip file.***

## Answer 3

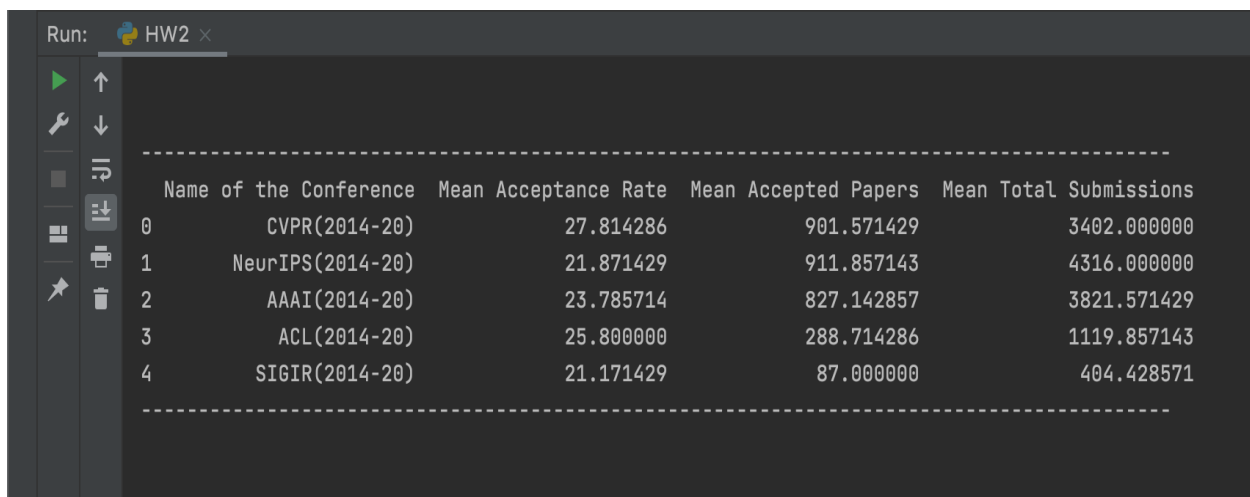
In this question we must analyze a given dataset which includes the statistics for paper acceptance rate in some of the main conference in the field of Artificial Intelligence. For this purpose, we are using various techniques to compare the visualizations using a table and a graph and compare and evaluate the results. I have chosen **multiple** cases to understand and solve this section which are all discussed below.

### Case 1

In this part I have used the entire table to calculate the **mean** of all the columns given in the dataset. This includes taking the mean of all the variables listed for each of the five different conferences.

1. Acceptance rate
2. Num. of accepted paper
3. Num. of total submissions

After taking the mean each of this datapoint is plotted on a scatter plot.

A screenshot of a Jupyter Notebook interface. The top bar shows 'Run: HW2 x'. On the left is a vertical toolbar with icons for running, undo, redo, and other notebook functions. The main area displays a table with the following data:

	Name of the Conference	Mean Acceptance Rate	Mean Accepted Papers	Mean Total Submissions
0	CVPR(2014-20)	27.814286	901.571429	3402.000000
1	NeurIPS(2014-20)	21.871429	911.857143	4316.000000
2	AAAI(2014-20)	23.785714	827.142857	3821.571429
3	ACL(2014-20)	25.800000	288.714286	1119.857143
4	SIGIR(2014-20)	21.171429	87.000000	404.428571

Figure 15. Case 1: Depiction using a Table

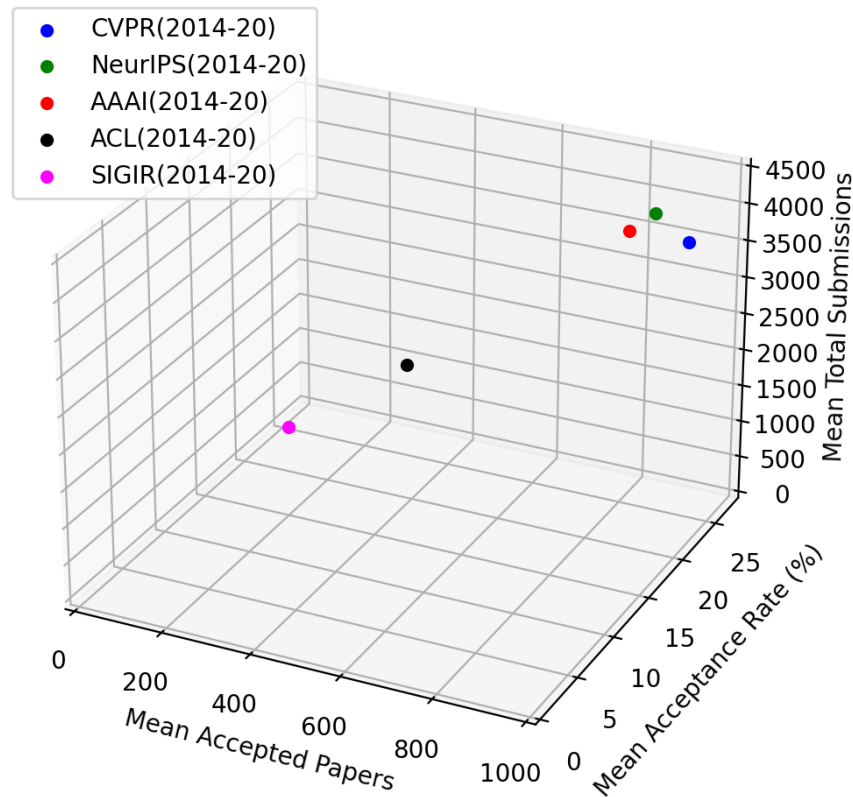


Figure 16. Case 1: Depiction using a scatter plot

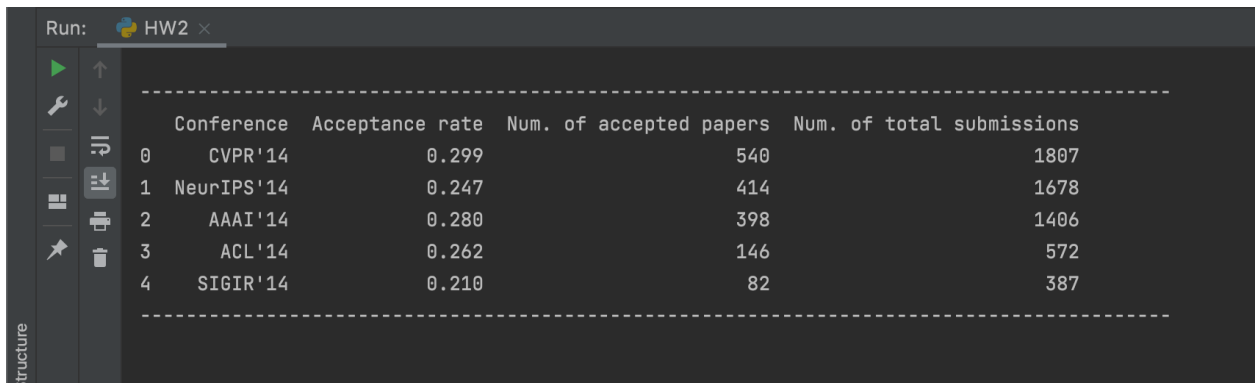
### Analysis for case 1

- In the first image we see a visualization using a table for the given dataset where we calculate the mean for each conference for every variable. The tabular form **looks clean and easy to read**. The reason behind this is since there is limited data we are working with.
- In the second image we see a visualization using a 3D scatter plot. The 3 variables are each shown on x, y and z-axis. This plot easily shows the relation between all these 5 conferences and how they match against each other. Although to know the **exact value** we need to be more **reliable on the tabular form**. Also, it can be difficult to understand such a plot like when we look at the Acceptance rate at an initial look, we feel that ACL has a lower mean conference rate as compared to NeurIPS which is not true as seen in the table.



## Case 2

In this part I have picked a particular year to carry on analysis of the 5 conferences. The dataset is trimmed to include only data from the year 2014 for each of the different conference. This data is then depicted using a bar plot as will be shown below.



The screenshot shows a Jupyter Notebook interface with a table of data for five conferences in 2014. The table has four columns: Conference, Acceptance rate, Num. of accepted papers, and Num. of total submissions. The data is as follows:

	Conference	Acceptance rate	Num. of accepted papers	Num. of total submissions
0	CVPR'14	0.299	540	1807
1	NeurIPS'14	0.247	414	1678
2	AAAI'14	0.280	398	1406
3	ACL'14	0.262	146	572
4	SIGIR'14	0.210	82	387

Figure 17. Case 2: Depiction using a table

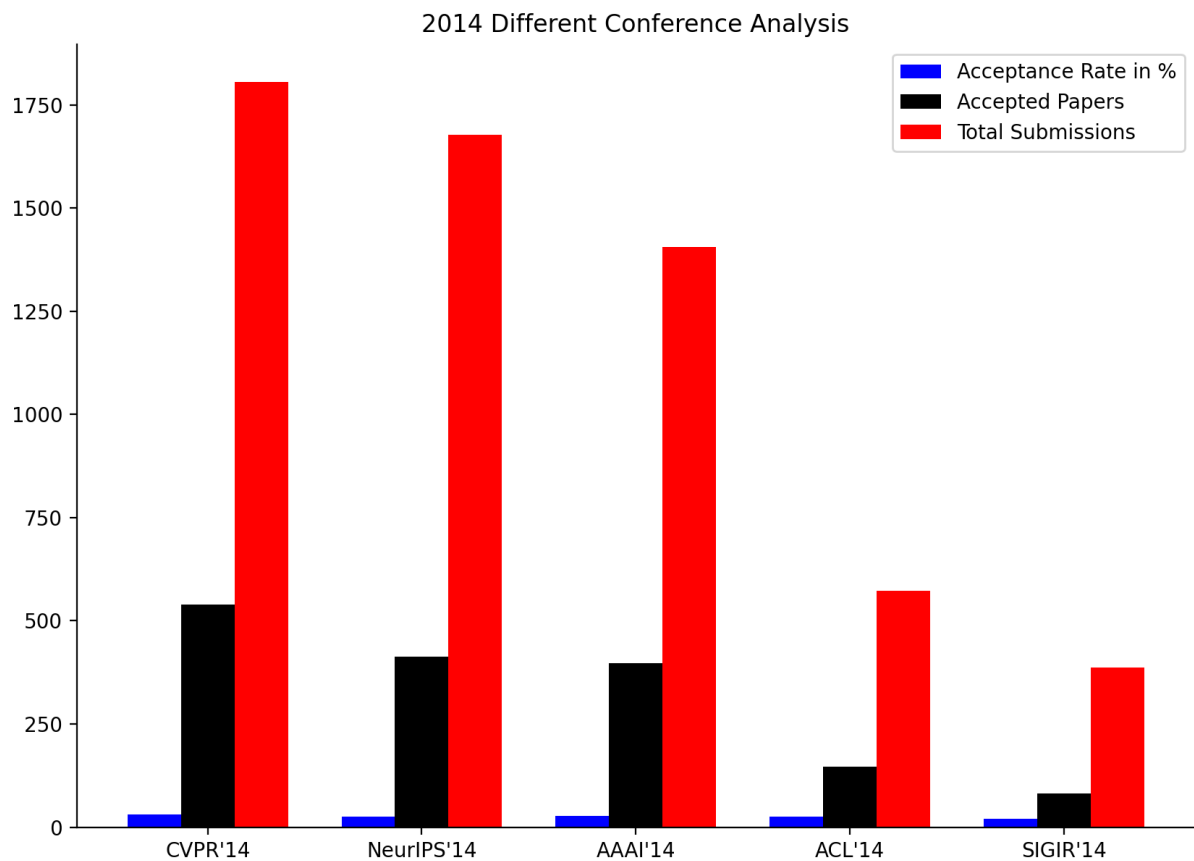


Figure 18. Case 2: Depiction using a bar plot

## Analysis for case 2

- In the first image we see a visualization using a table where the original dataset is trimmed to show only the **cases of conference from the year 2014**. The data is small and efficient which makes it easy to read and interpret.
- In the second image we see a visualization using a bar plot. This plot shows us the **comparison** between these conferences more easily as we can judge or estimate the quantity using the height of each bar they depict. The bar plot shows the data more efficiently as compared to the scatter plot. One disadvantage of this it that if we look at the plot, we can see that it is **hard** to tell the **difference** between Acceptance rate among different conferences since the magnitude of scale on y-axis is too large. To overcome this, we can update the plot to one shown below.

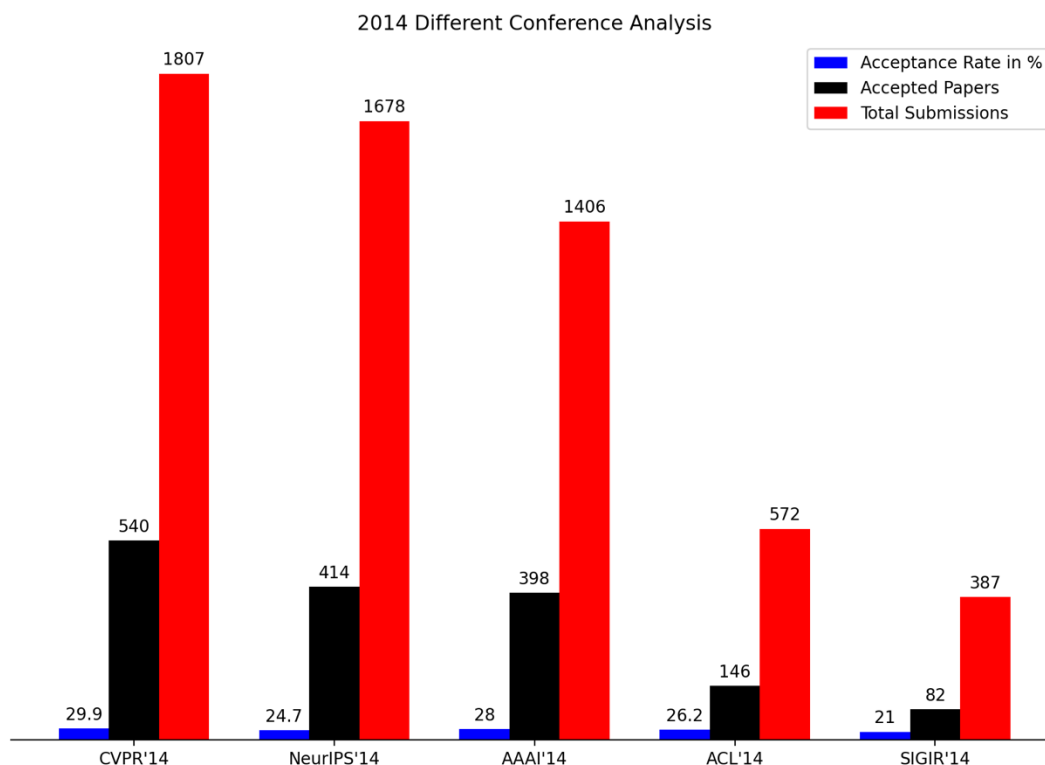


Figure 19. Updated depiction using a bar plot

This plot eliminates the y-axis as whole and instead uses data labels above each plot to show the values of that particular variable. This helps us in easy understanding and more convenient reading especially for the acceptance rate.

### Case 3

In this section I have used the entire dataset. No modification was made to the original dataset. In the plot a distribution chart is shown for each of the variable that shows us they are spanned over the entire dataset.

	A	B	C	D	E
1	Conference	Acceptance rate	Num. of accepted papers	Num. of total submissions	
2	CVPR'14	29.90%	540	1807	
3	CVPR'15	28.30%	602	2123	
4	CVPR'16	29.90%	643	2145	
5	CVPR'17	29.90%	783	2620	
6	CVPR'18	29.60%	979	3303	
7	CVPR'19	25.00%	1294	5160	
8	CVPR'20	22.10%	1470	6656	
9	NeurIPS'14	24.70%	414	1678	
10	NeurIPS'15	21.90%	403	1838	
11	NeurIPS'16	23.60%	549	2403	
12	NeurIPS'17	20.90%	678	3240	
13	NeurIPS'18	20.80%	1011	4856	
14	NeurIPS'19	21.10%	1428	6743	
15	NeurIPS'20	20.10%	1900	9454	
16	AAAI'14	28.00%	398	1406	
17	AAAI'15	26.70%	531	1991	
18	AAAI'16	25.80%	549	2132	
19	AAAI'17	24.60%	638	2590	
20	AAAI'18	24.60%	933	3800	
21	AAAI'19	16.20%	1150	7095	
22	AAAI'20	20.60%	1591	7737	
23	ACL'14	26.20%	146	572	
24	ACL'15	25.00%	173	692	
25	ACL'16	28.00%	231	825	
26	ACL'17	25.00%	195	751	
27	ACL'18	25.30%	258	1018	
28	ACL'19	25.70%	447	1737	
29	ACL'20	25.40%	571	2244	
30	SIGIR'14	21.00%	82	387	
31	SIGIR'15	20.00%	70	351	
32	SIGIR'16	18.00%	62	341	
33	SIGIR'17	22.00%	78	362	
34	SIGIR'18	21.00%	86	409	
35	SIGIR'19	19.70%	84	426	
36	SIGIR'20	26.50%	147	555	
37					

Figure 20. Depiction using a table

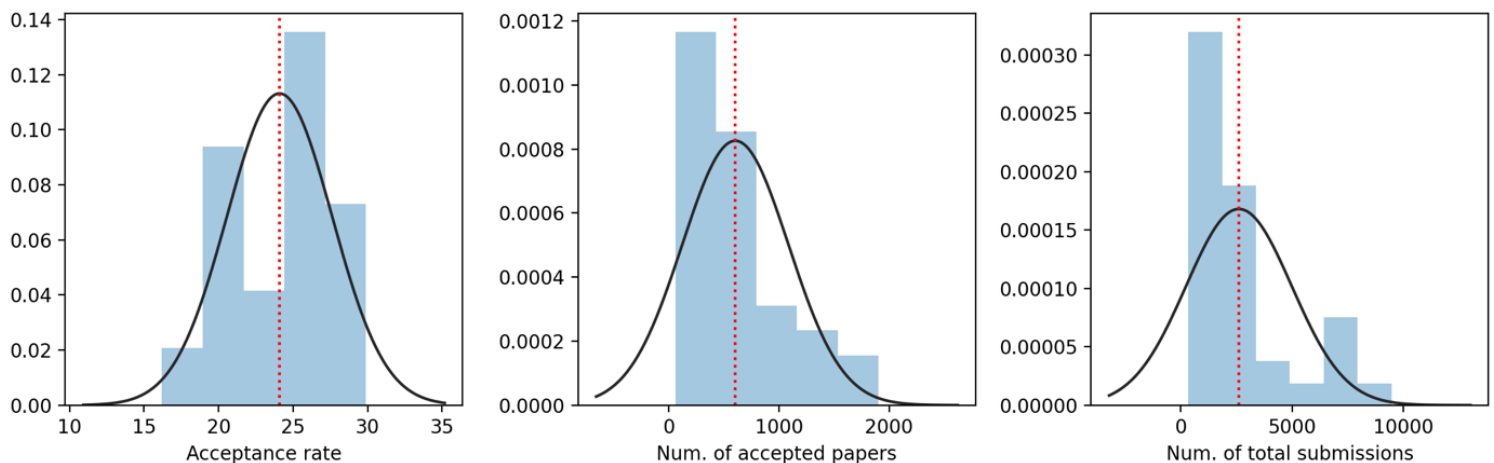


Figure 21. Depiction using a distribution chart

### **Analysis for case 3**

- In the first image we see a visualization using a table. This is the input dataset as given in the question **without any modifications**. This is considerably **large** to analyze at a same time. So, plotting some sort of graph would be useful.
- In the second image we see a visualization using a distribution plot for all the 3 variables. This helps us understand the **variance** in the entire data while telling us about the **minimum, maximum and the mean** values all at the same time without really looking into the entire table. The plot for acceptance rate is quite **evenly** distributed with a mean value of approximately 24%. The plots for Num. of accepted papers and the Num. of total submission are both **right skewed** meaning that there are greater number of values below the mean in each case.

### **Case 4**

In this last section for this question, I am trying to analyze a particular conference over the years from 2014-2020 using a table and a subsequent line plot as shown below.

Run: HW2 x

	Conference	Acceptance rate	Num. of accepted papers	Num. of total submissions
0	AAAI'14	0.280	398	1406
1	AAAI'15	0.267	531	1991
2	AAAI'16	0.258	549	2132
3	AAAI'17	0.246	638	2590
4	AAAI'18	0.246	933	3800
5	AAAI'19	0.162	1150	7095
6	AAAI'20	0.206	1591	7737

Figure 22. Depiction using a table

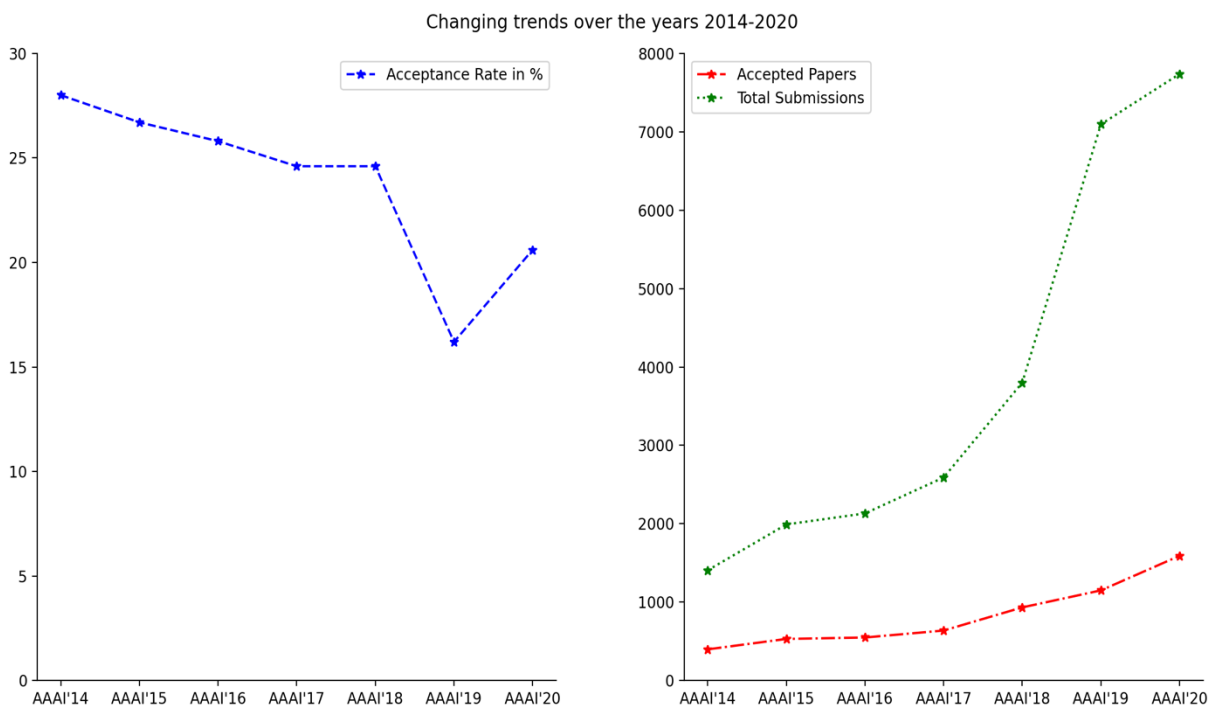


Figure 23. Depiction using a line chart

### Analysis for case 4

- In the first image we see a visualization using a table which is obtained by trimming the original dataset to contain only the cases for the AAAI conference over the years.
- In the second image we see a visualization using a line chart. I would say this is more **superior as compared to the tabular form** since it shows us the **changing trend** over the years for all the variables. Two subplots are shown to have a clear picture as plotting the acceptance rate along with the other two variables would make it difficult to visualize it since the magnitude on y-axis is quite large compared to the span for acceptance rate. We can easily see the **ups and downs** over the years. A similar plot showing changing trends for all conferences can also be plotted.

*In all it can be concluded that both tabular form of depiction and use of some sort of graphs are efficient methods to visualize data. Depending on what we need to interpret and how big our dataset it there are pros and cons related to them based on case-to-case basis as discussed above. Tabular depiction in general provides us with precise values whereas plots give us about the changing trends and comparisons between various cases and variables.*

## Answer 4

At a first glance, for any ordinary person it is easy to spot the difference and the similarities between the two given images. In naïve terms we can say that both images have a similar background with a splash of yellow color. The difference between them is that one pictures show alphabets whereas the other one has numbers in it. However, we need to understand and read between these images. We need to find a hidden aspect to what is similar and what is different.

Talking in data visualization terms we need to visualize them using concepts of **perception and cognition**. In both the images the middle element depicts the **exact same thing**. The difference is the surrounding elements which is bringing the **perceptive** analogy in play.

In the first image we have a list of alphabets A, B and C and in the second image we have the numbers 12, 13 and 14. The alphabet B and the number 13 are written in the same manner. Our human brain is trained in such a manner as it **perceives the data on the surrounding to identify and label any element**. The similarity and difference both lie around the middle part in the two images.

So, for the first image looking at surrounding alphabets we think that it is also an alphabet, so we label it as B. Whereas in the second image we have numbers on either side which makes us think that the middle element is also a number and hence we end up seeing it as 13. This is the same way as a group of dots designed in such a way that the **spacing** between them decides whether they denote a **row** of dots or a **column** of dots. Human brain decides the marking as a number, or an alphabet based on the **knowledge** of the elements on either side. Our initial perception is based on the image we see first based on the way we **selectively interpret** the data. So, to conclude we can say that depending on **the context of data** we can identify the middle element as B or 13.