**Software Engineering Tools Lab**

**Assignment No-1**

**(Module 1- Introduction to OSS)**

**Name: Avinash Vishnu Biradar**

**PRN: 2019BTECS00077**

**Batch: T7**

**-------------------------------------------------------------------------------------------------------------**

**Q1. Using Weka tool perform below tasks such as data preprocessing, data classification (use any appropriate ML algorithm) and data visualization efficiently on given dataset.**

🡪

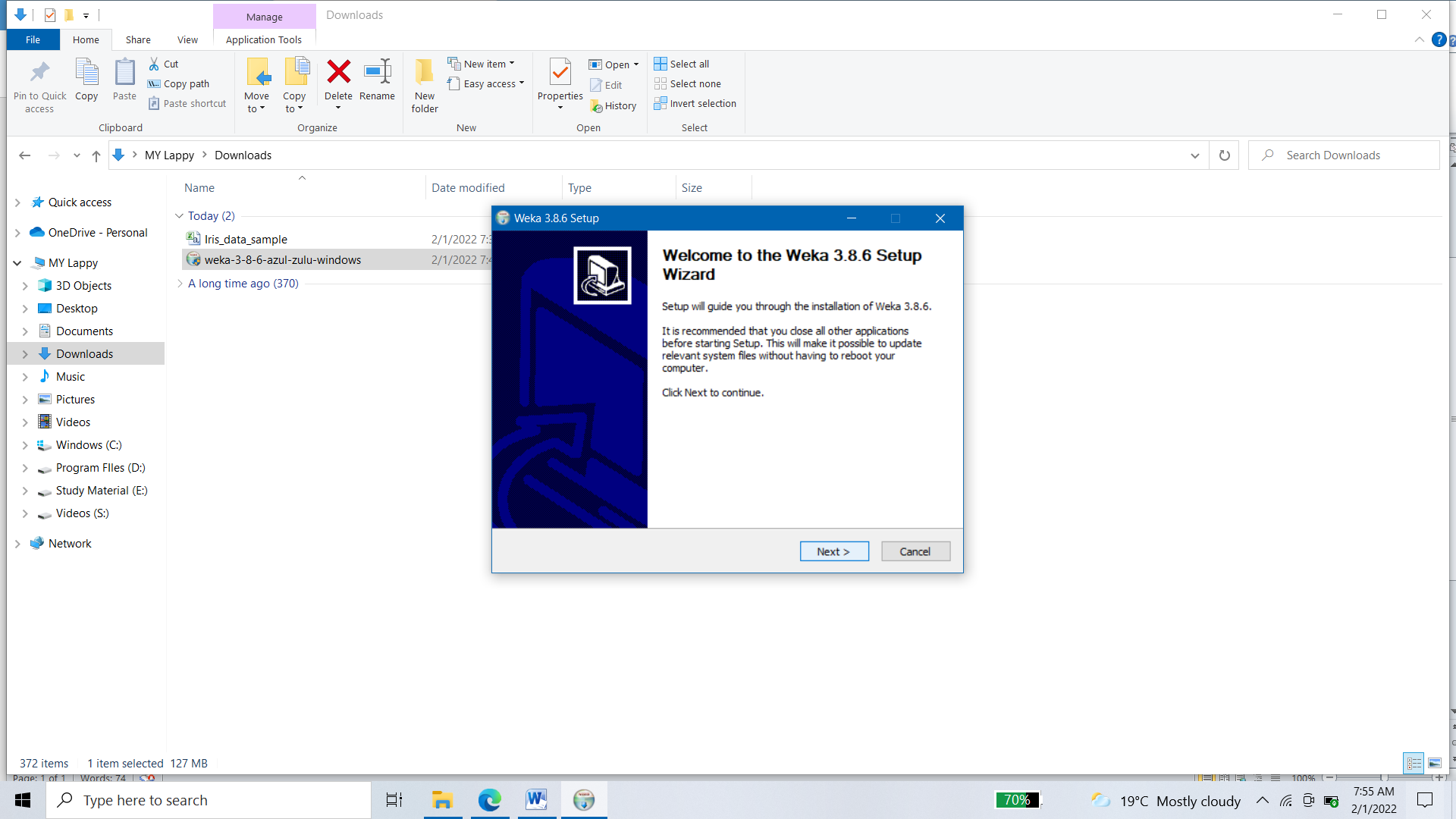
Data visualization using weka

I have used Iris.csv as provided .

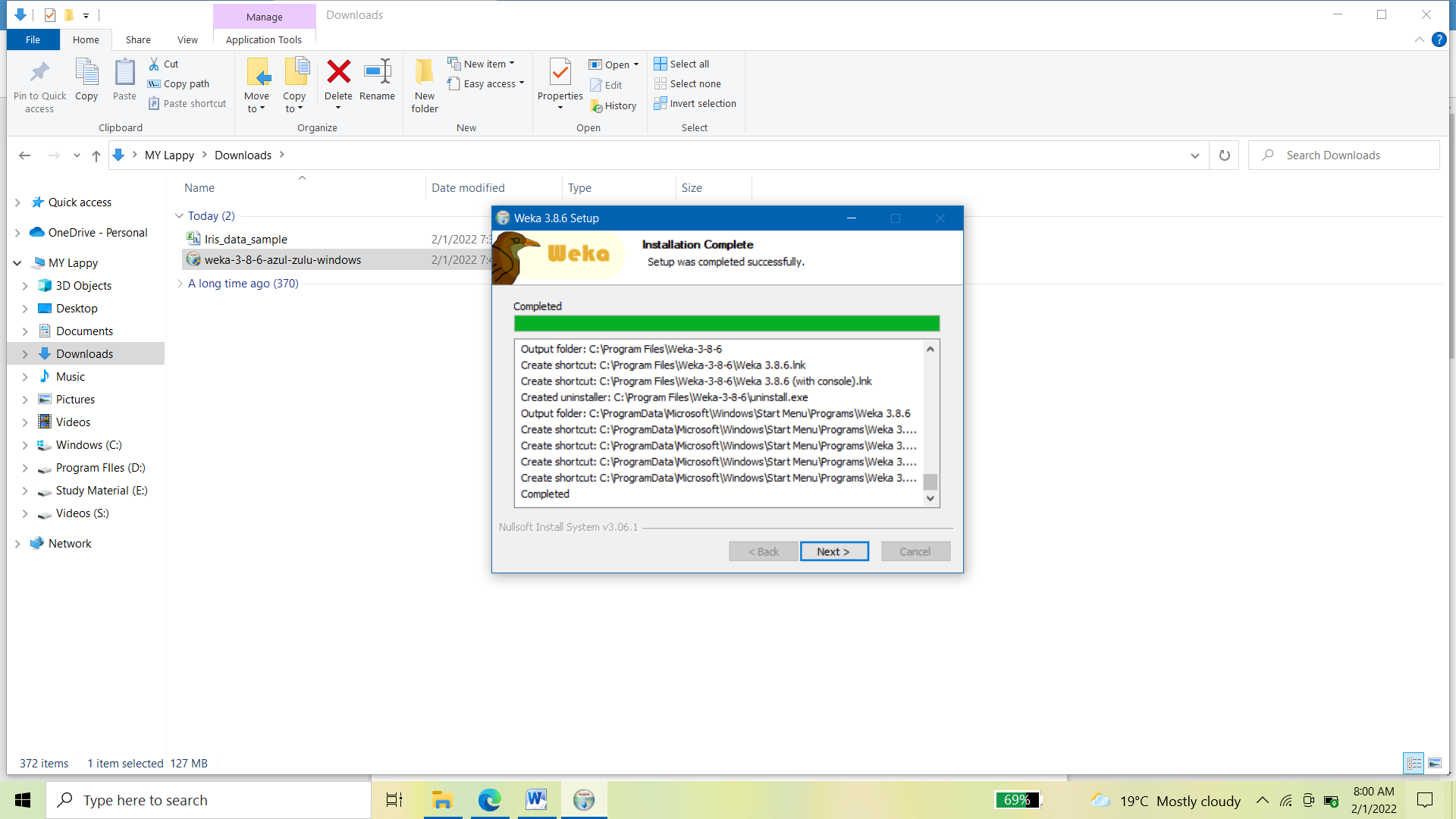
After importing I have used various parameters or fields to represent their frequency, missing values, uniqueness and NAN values.

**Step 1: Downloading Weka and loading data file,**

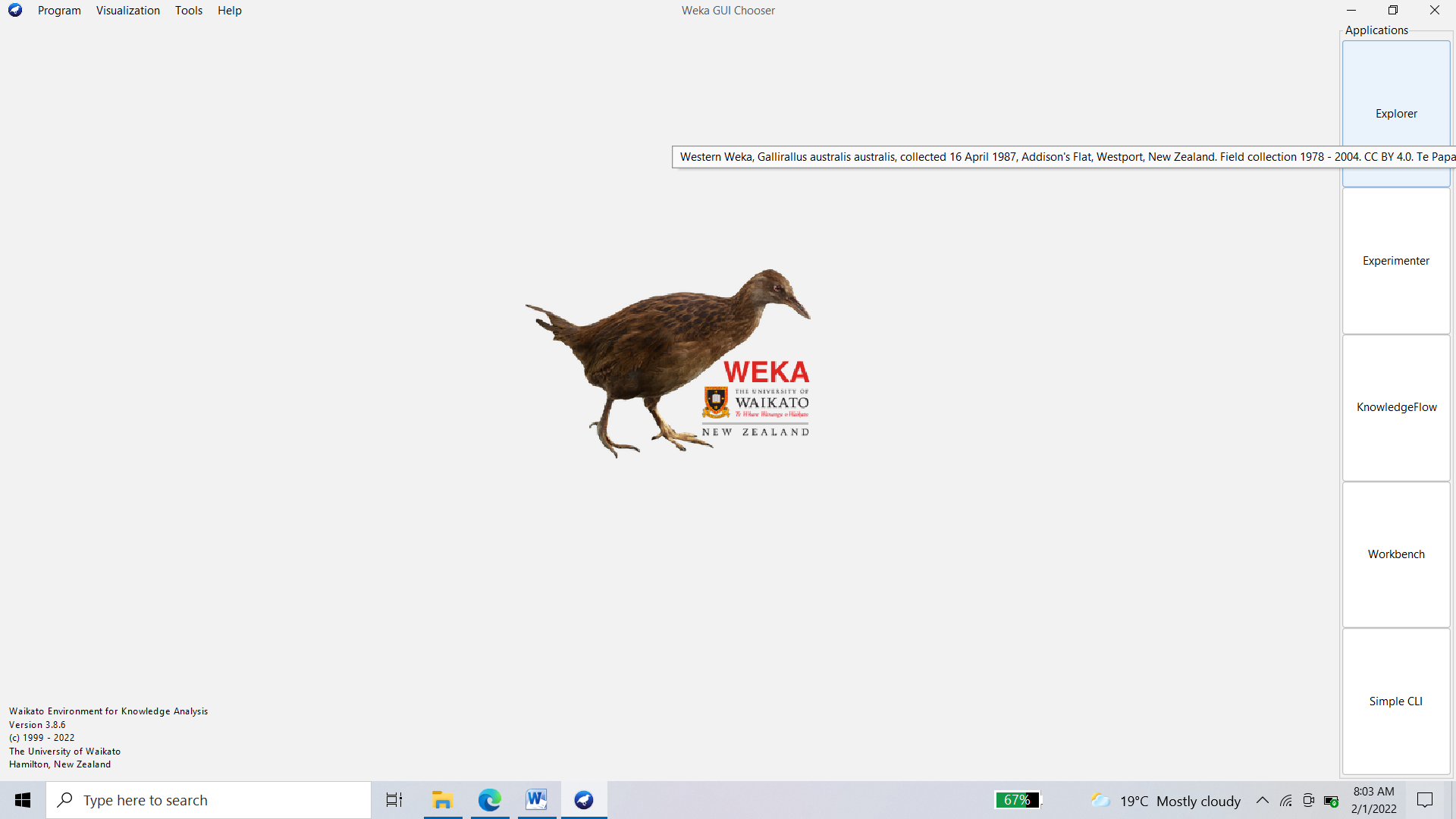
1. Visited the official download page of weka, on clicking the windows download it will redirect to open-forge website and then clicked the download button. Weka application file has downloaded.



1. Clicking on next, Proceeded with default settings, i.e. Full Associate, and finally click the install button.

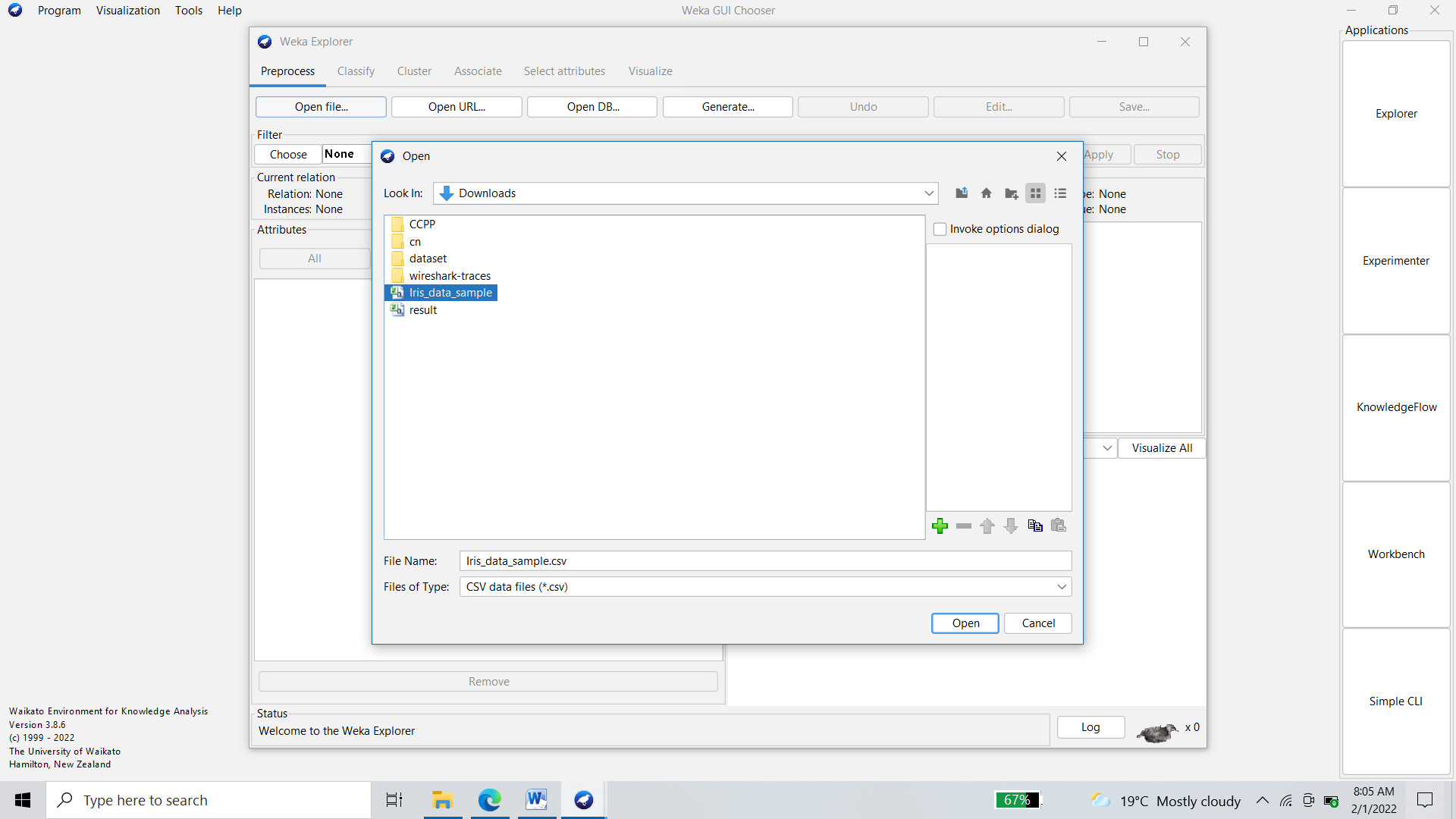


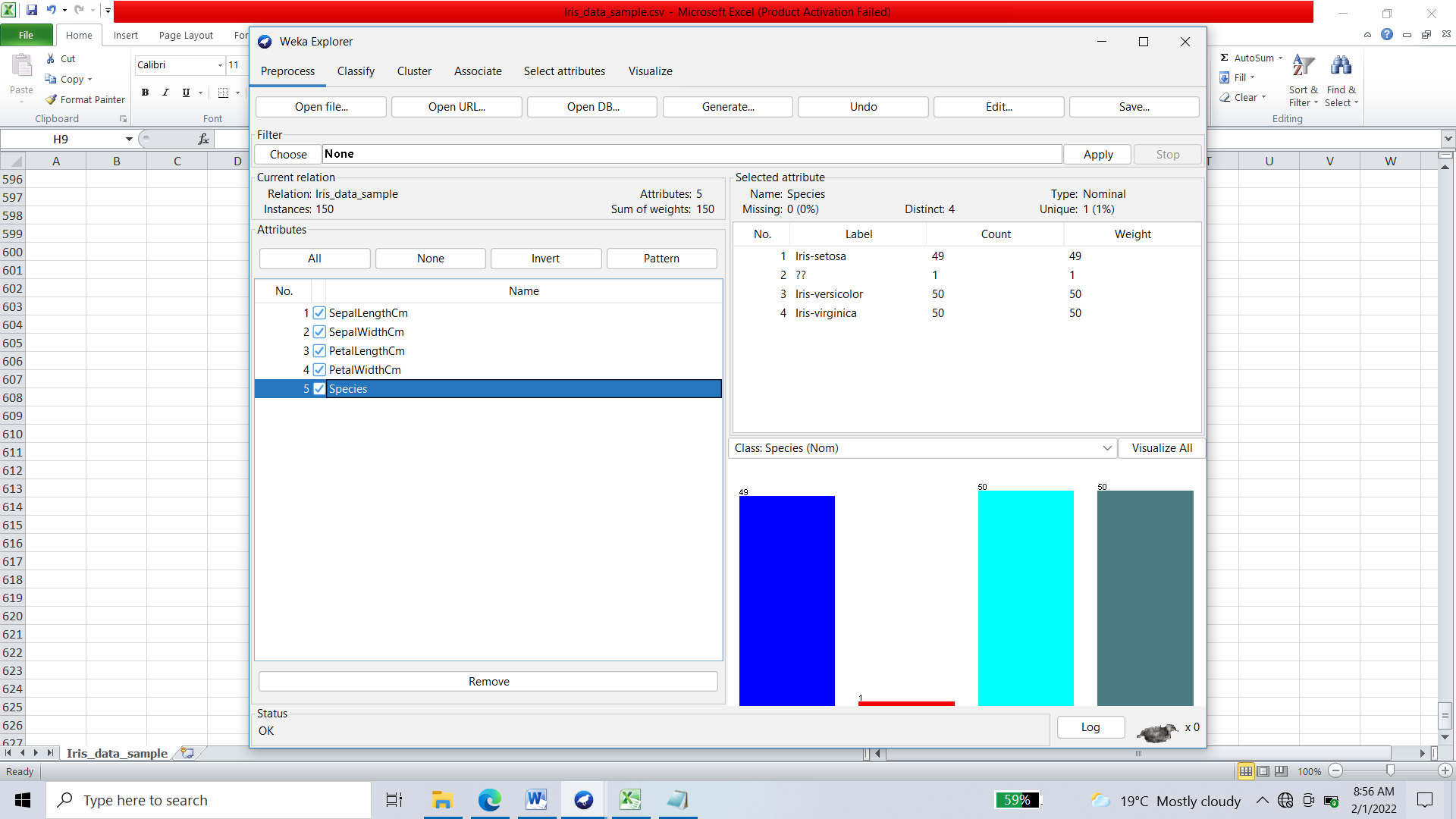
1. Clicked the next -> finish and open the software from start menu.



Clicked the explorer tab and selected open file to choose iris dataset

that is downloaded.

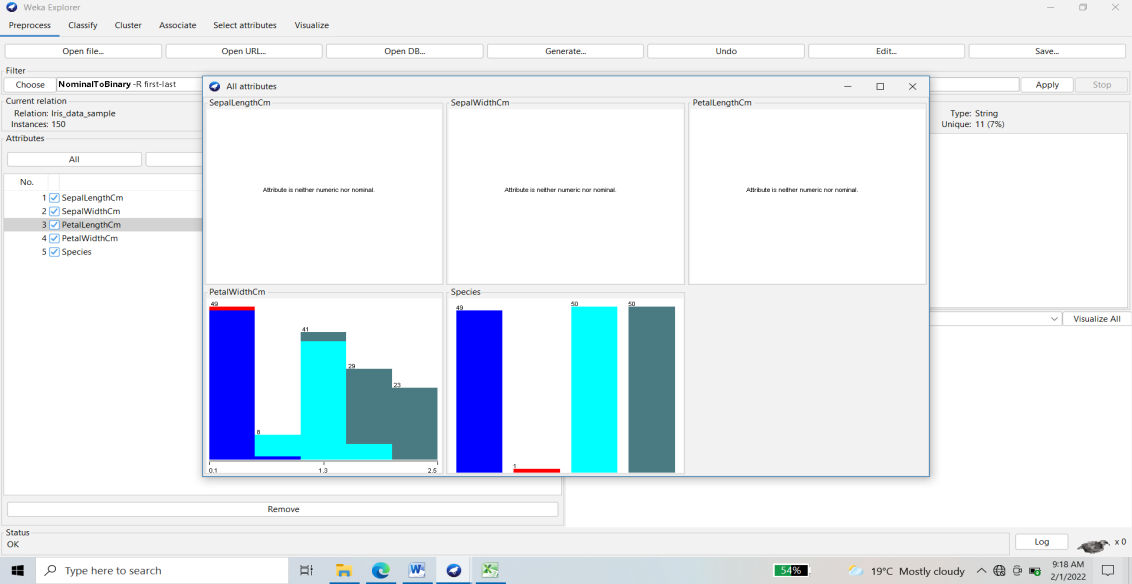




After deleting the first column of index, and selecting the dataset, we get the following with some visualization of count of each element.

**Step 2: Data Preprocessing**

1. On selecting the visualize all ,



1. To get first three attributes visualization, click the choose before the filter, select the **Weka ->Filter->unsupervised->attributes->StringToNominal**

Select the attribute which needs to change from string to nominal

And change attributes range by clicking on filter name and click apply.

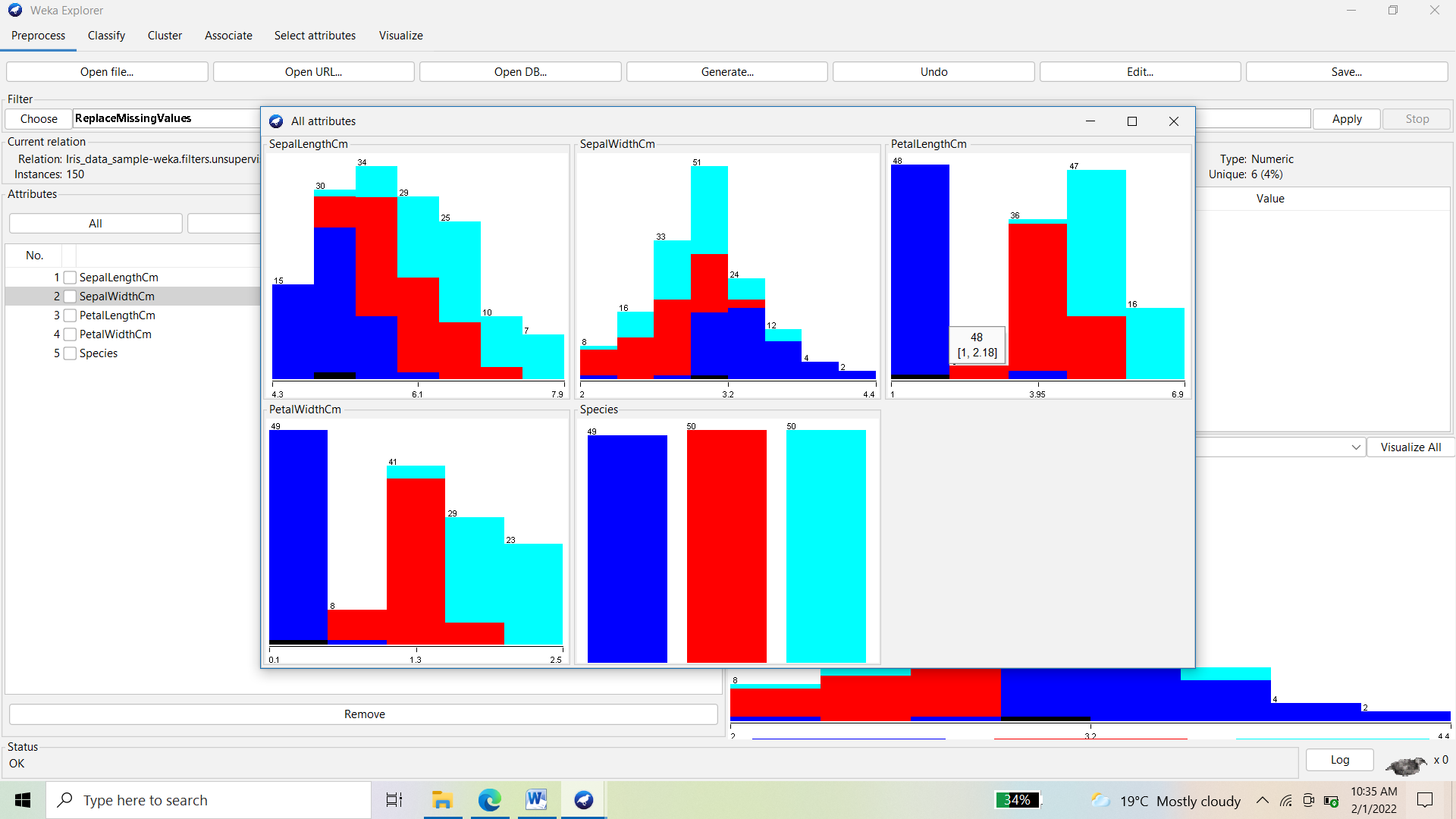
OR

Open the dataset in MS excel and change the ?? with ?, as ? specifies the missing element, and #### with NULL. And now save the updated file. And open the updated file in Weka.

You can see that string has been change to numeric data.

Choose visualize all.

1. Remove the null values by finding Replace missing values in filter of unsupervised filter.



Step 3: Classification using naïve bayes and others

1. Select the classify tab from toolbar,
2. Select the percentage split as 50%
3. Choose the classification method as naïve bayes
4. Select the (Nom) species
5. Click start.

We get the accuracy with naïve bayes as 96%.

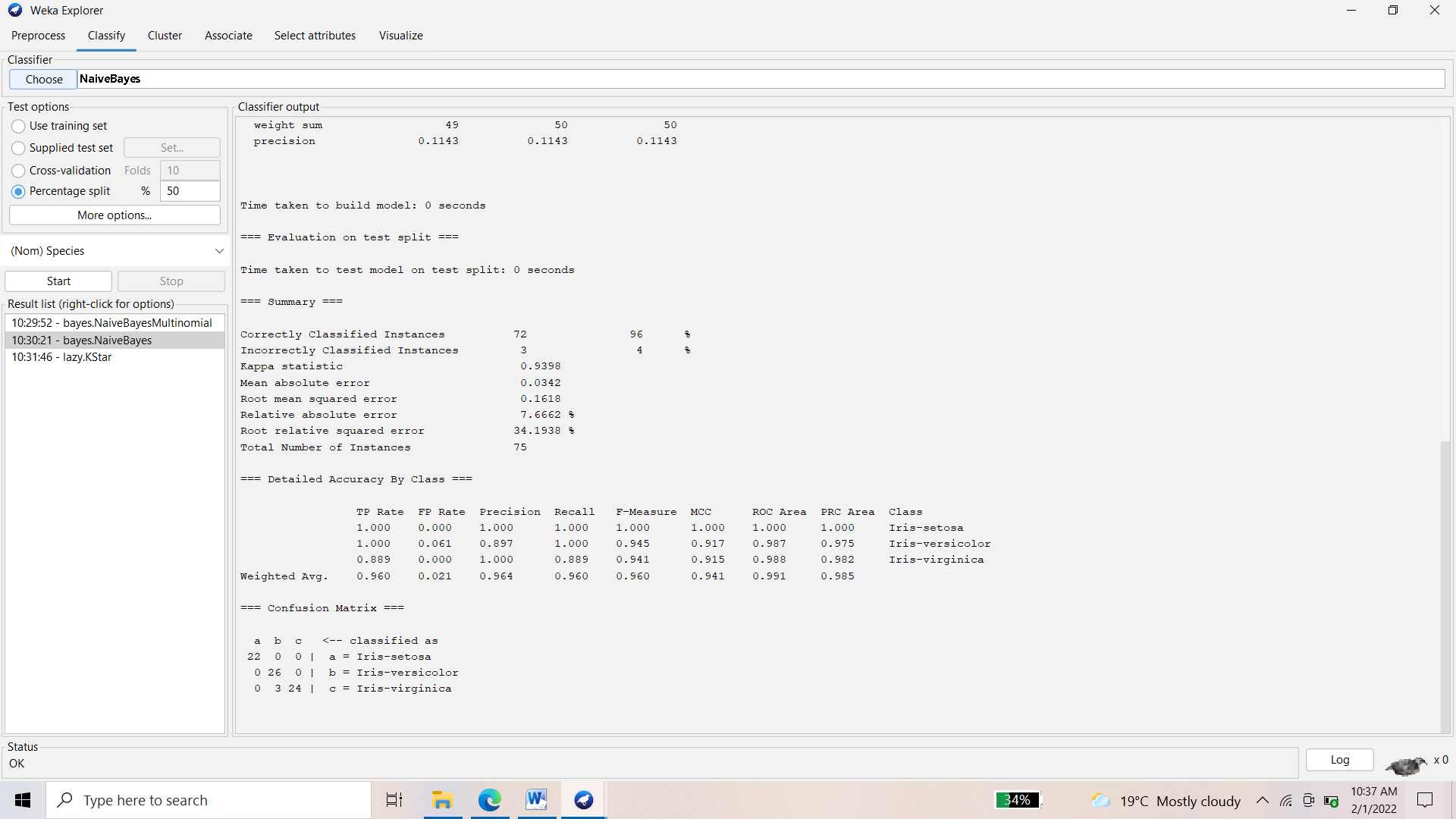
=== Confusion Matrix ===

a b c <-- classified as

22 0 0 | a = Iris-setosa

0 26 0 | b = Iris-versicolor

0 3 24 | c = Iris-virginica



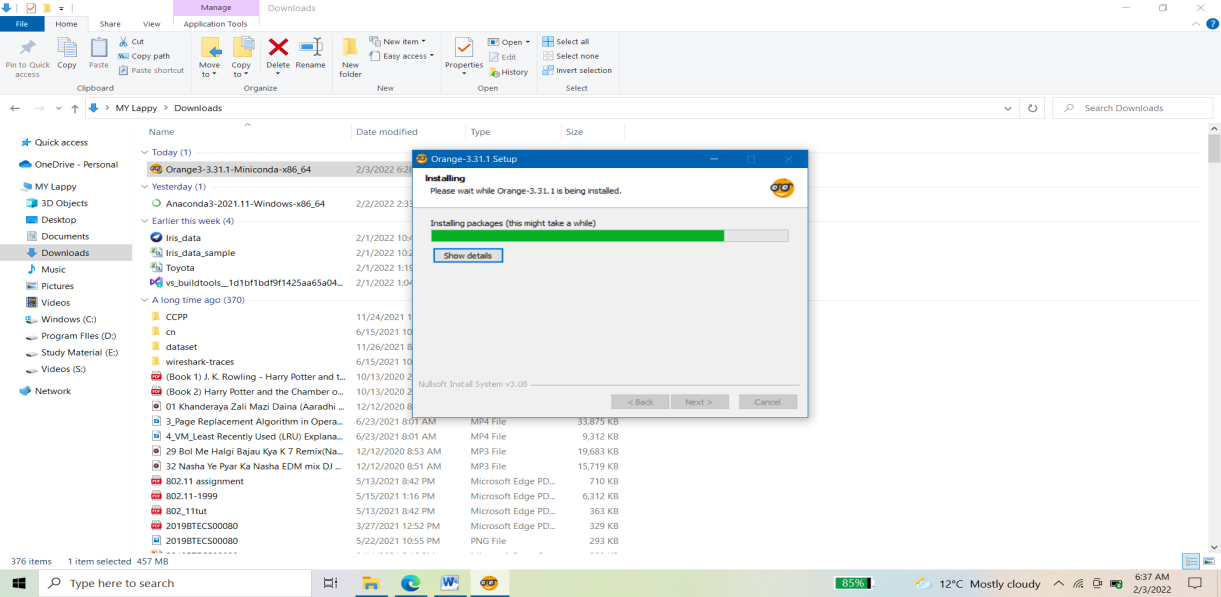
**Q2. a. Install orange**

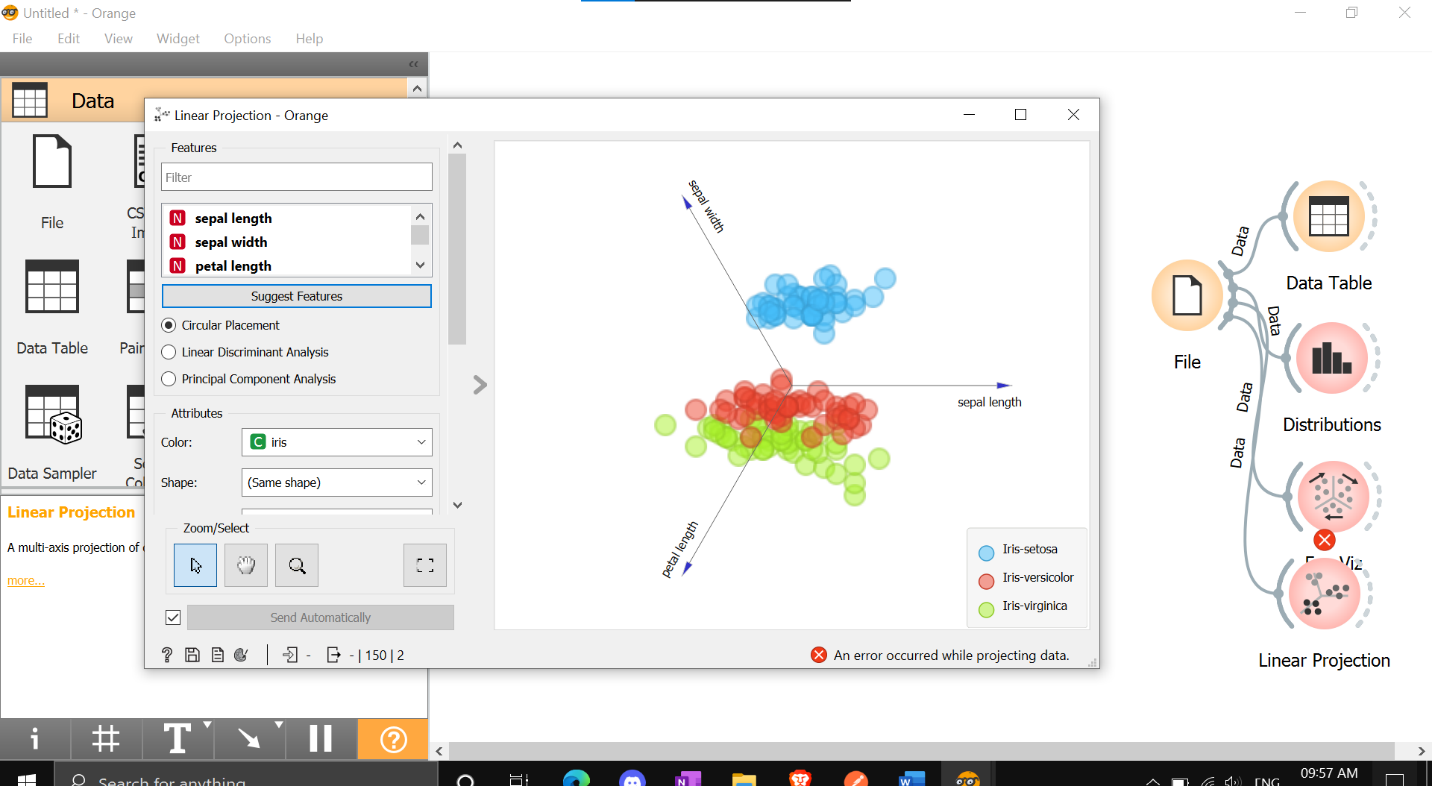
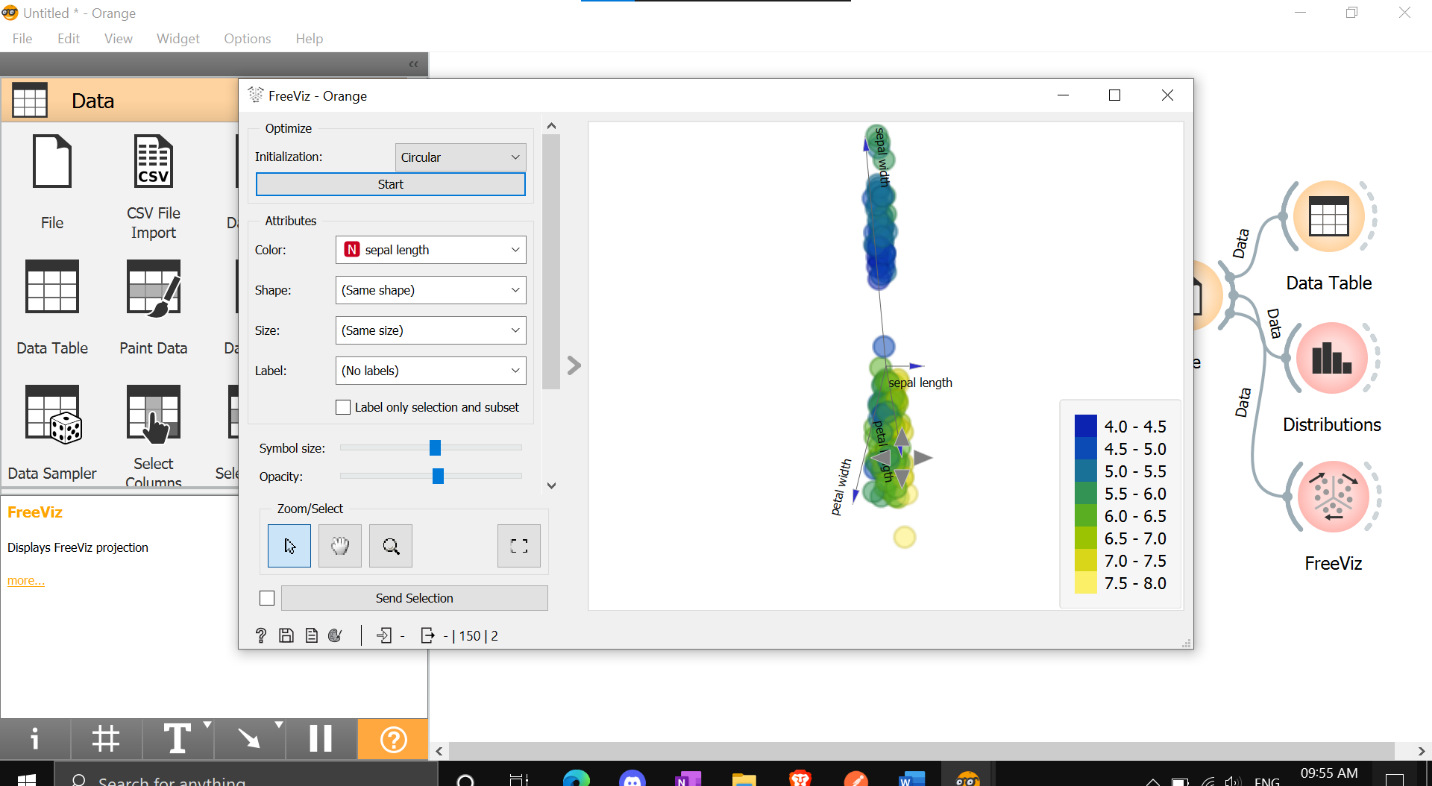
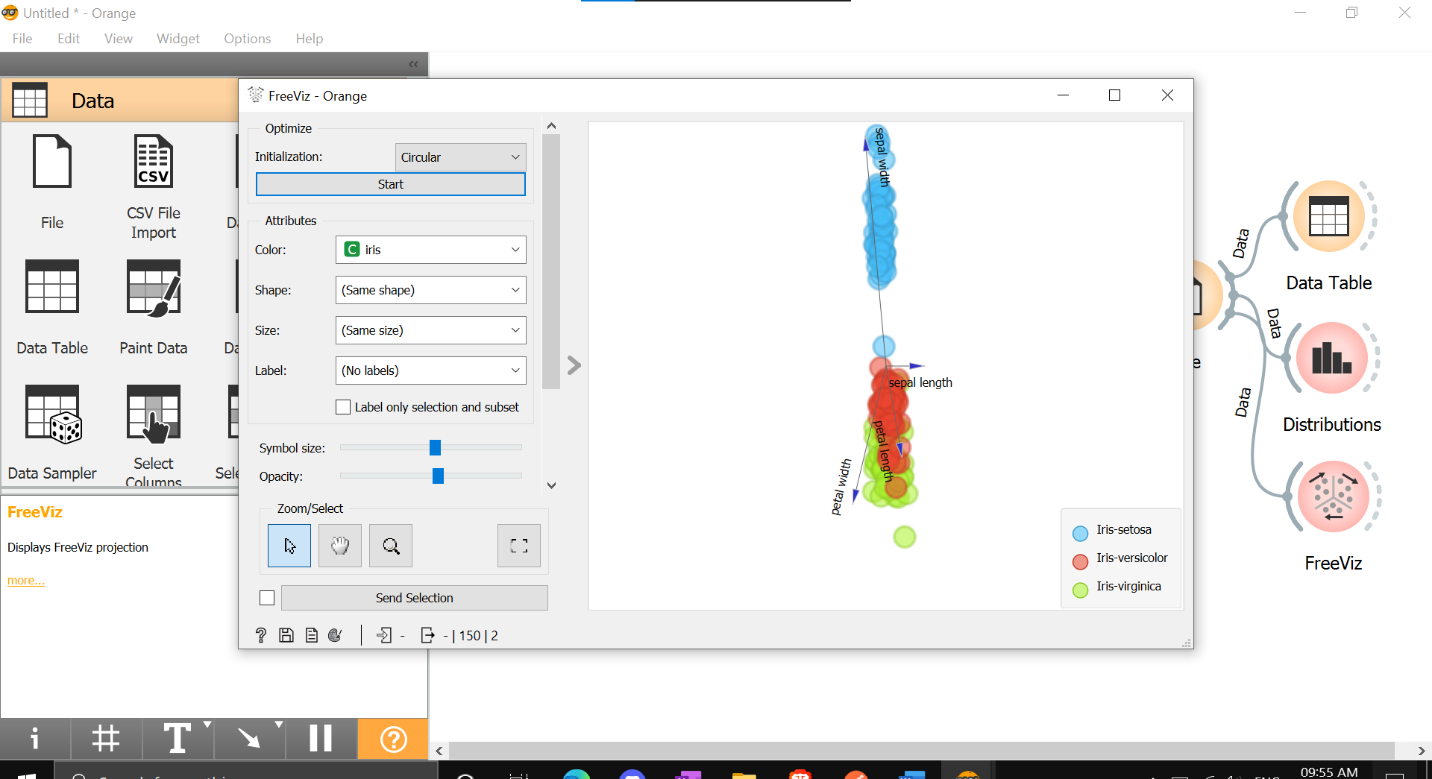
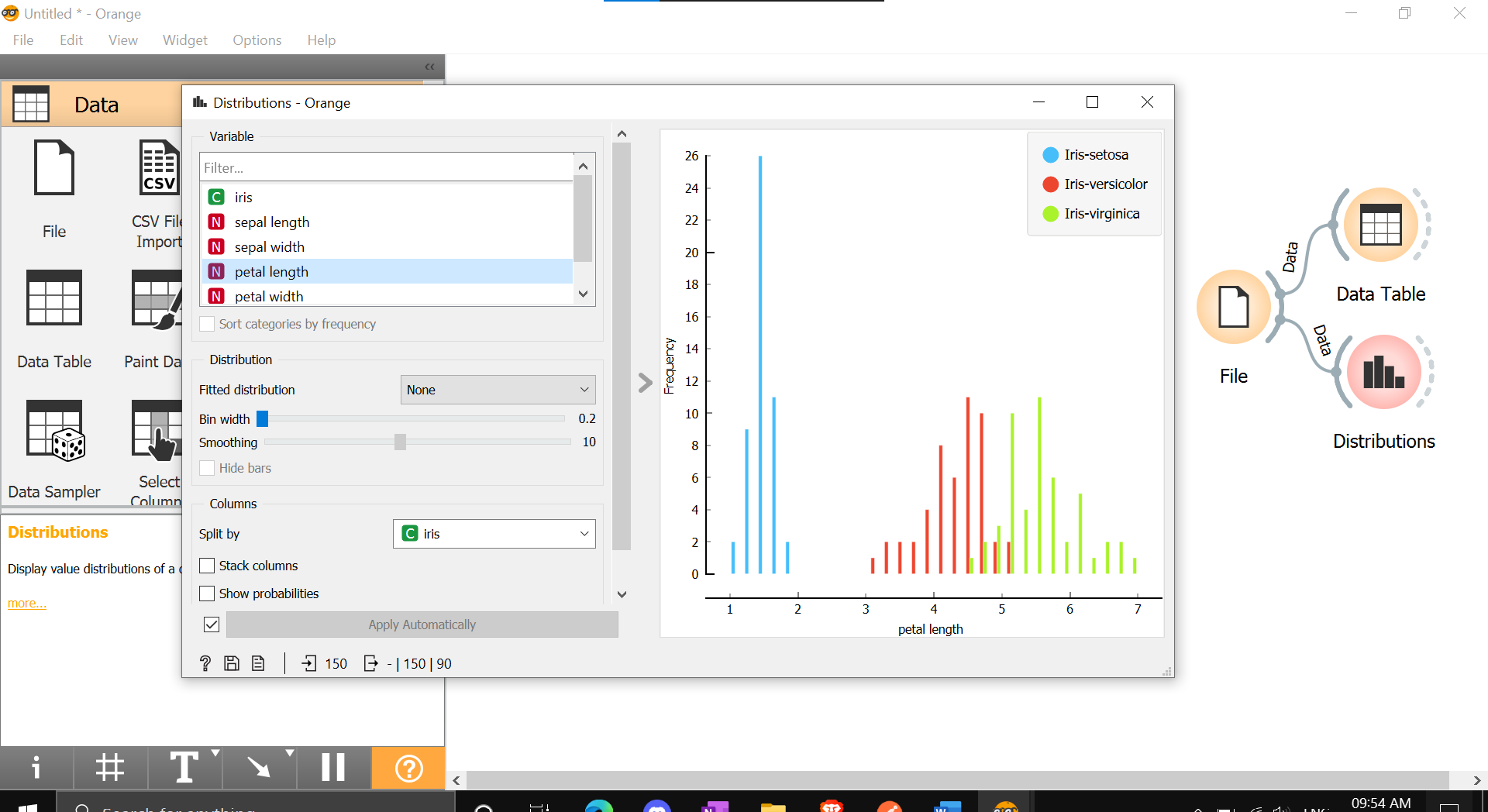
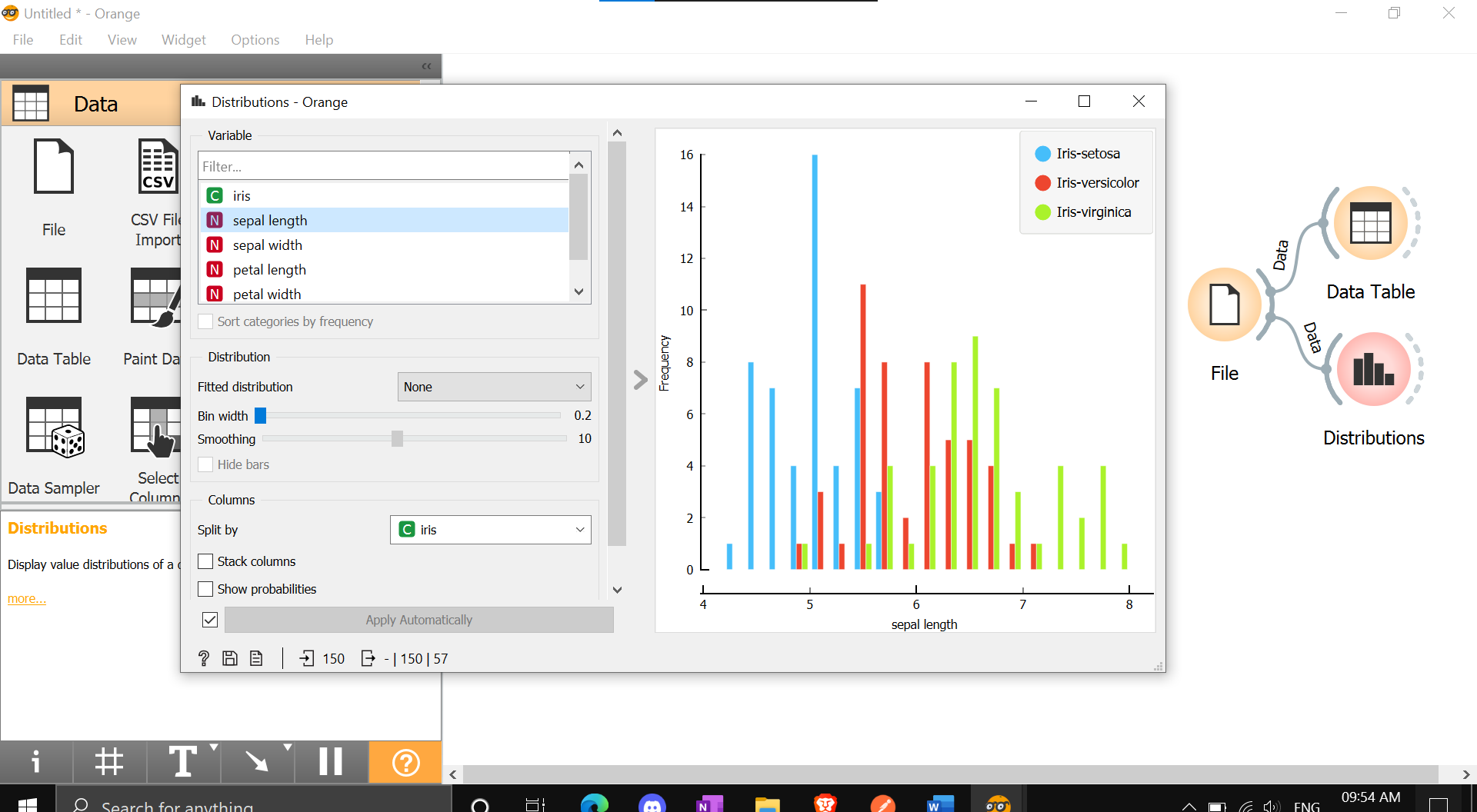
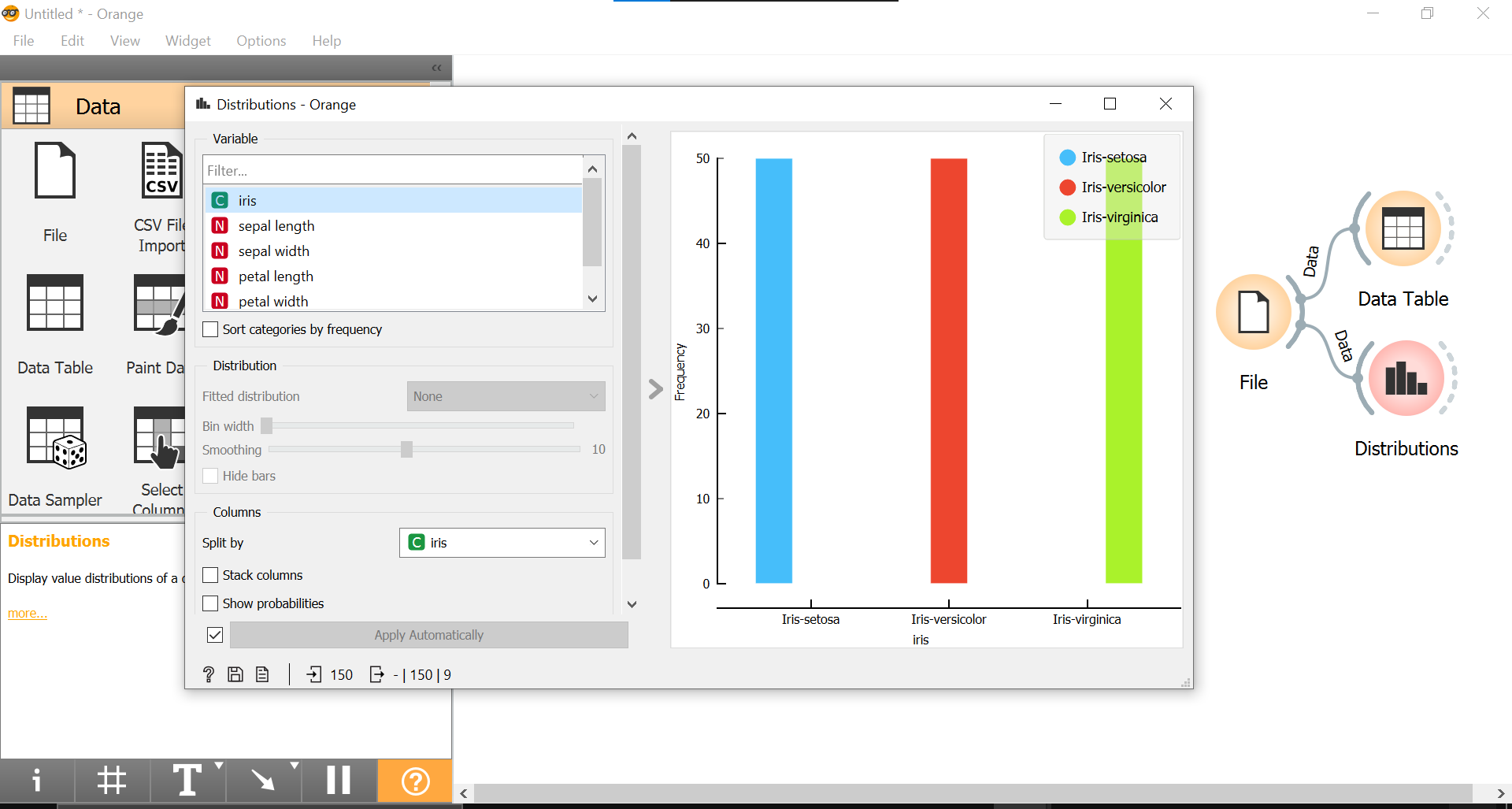
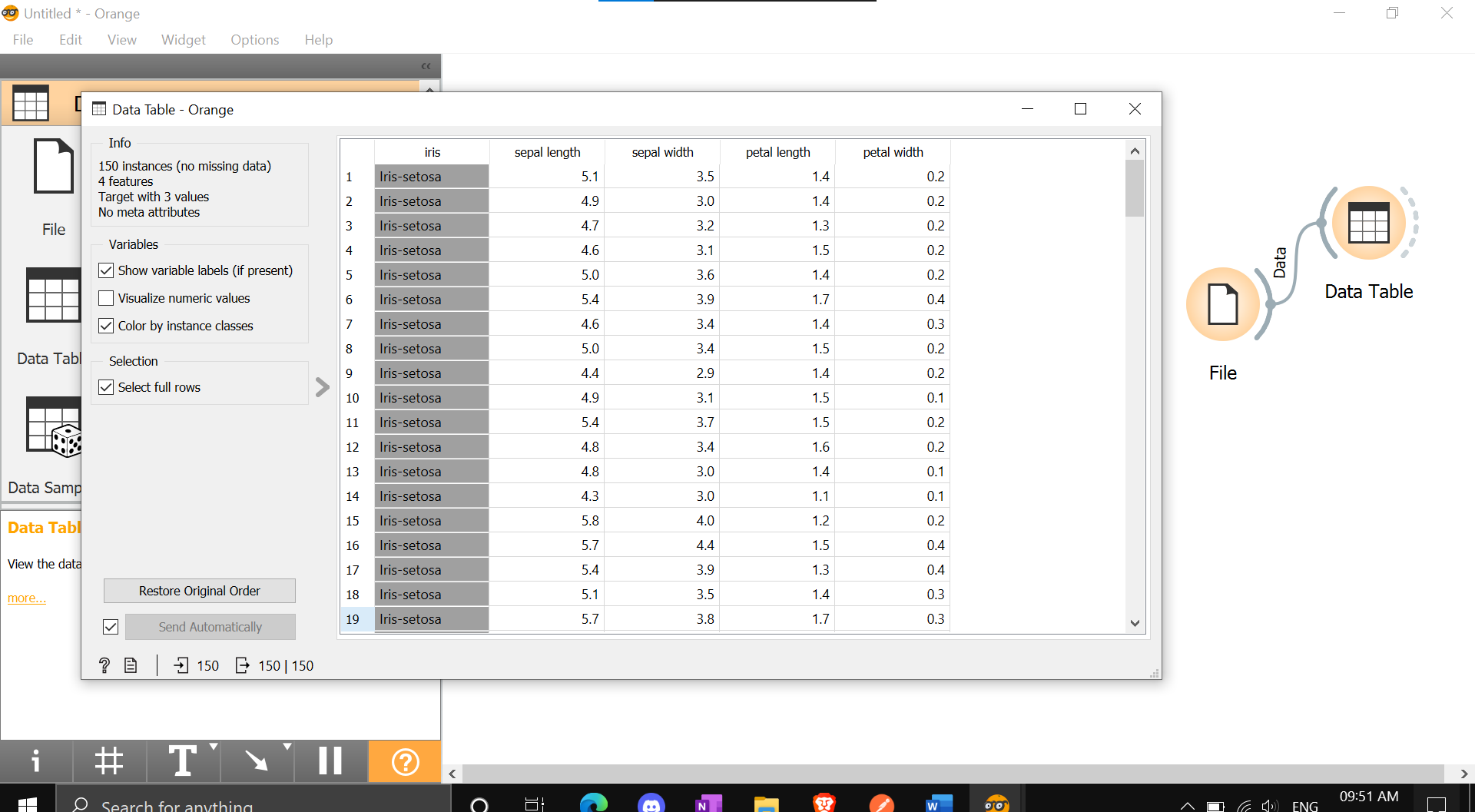
**b. Show data distribution**

**c. Show linear projection**

**d. Show FreeViz**

**🡪**

****

****

**Q3.** Differentiate in between free software, Open source software and proprietary software with respect to its properties.

🡪

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter of Comparison | Free software | Open source Software | Proprietary Software |
| Definition | “Free software” means software that respects users’ freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. | Open Source software refers to software that contains a source code with license which is open to free use, modification and distribution. | Proprietary software refers to the type of software that contains a licensed source code and is copyrighted for use. |
| Collaboration | The term “free software” is sometimes misunderstood—it has nothing to do  with price. It is  about freedom | Open Source software is developed for open collaboration. | Proprietary software is not meant for open collaboration, but only for the creator and users who have paid for it. |
| Access | Software freedom translates to social freedom. | Open-Source software has open access, that is, can be accessed by anyone. | Proprietary software can be accessed only by those who developed it and those who have paid for it. |
| Flexibility | Free Software  Is most Flexible, it can used by anyone. | Open Source  software is flexible, that is, it can be used, modified and | Proprietary software has restricted flexibility, that is,  there are |
|  |  | distributed by anyone | restrictions on its usage. |
| Examples | The Free Software Directory maintains a large database of free-software packages. Some of the best-known examples include the Linux kernel, the BSD and Linux operating systems, the GNU Compiler Collection and C library; the MySQL  relational database; the Apache web server; and the Send mail transport agent. | FreeBSD (Berklee  Software  Distribution),  Android,  LibreOffice, [Ubuntu](https://askanydifference.com/difference-between-fedora-and-ubuntu/)  are a few examples of Open Source software. | Windows, Microsoft, macOS, [Adobe Photoshop,](https://askanydifference.com/difference-between-apple-aperture-and-adobe-photoshop/) Adobe Flash Player are a few examples of Proprietary software. |

Q4)Using Anaconda Python create Histogram, Scatter plot and Bar plot for the dataset given below.

1. Scatter plot- Scatter plot of Price Vs Age
2. Histogram- for KM and CC
3. Bar plot- Bar plot for different fuel types

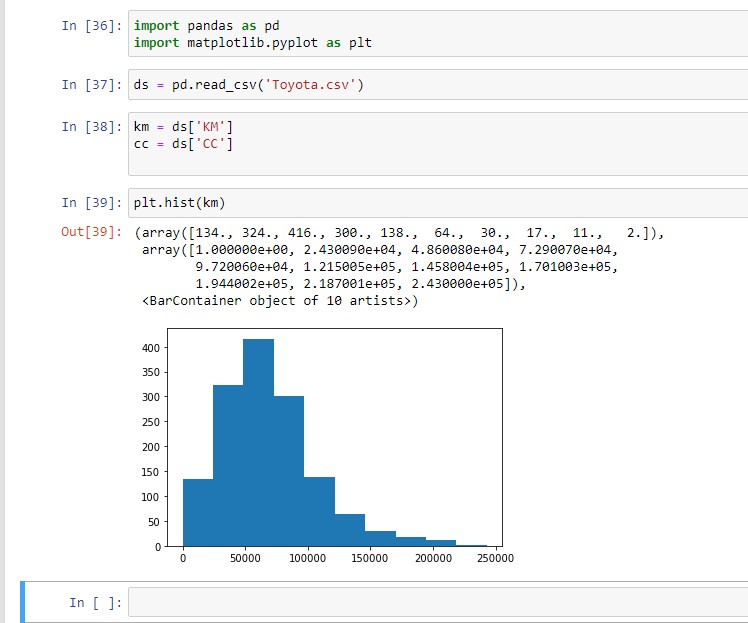
## Anaconda Python

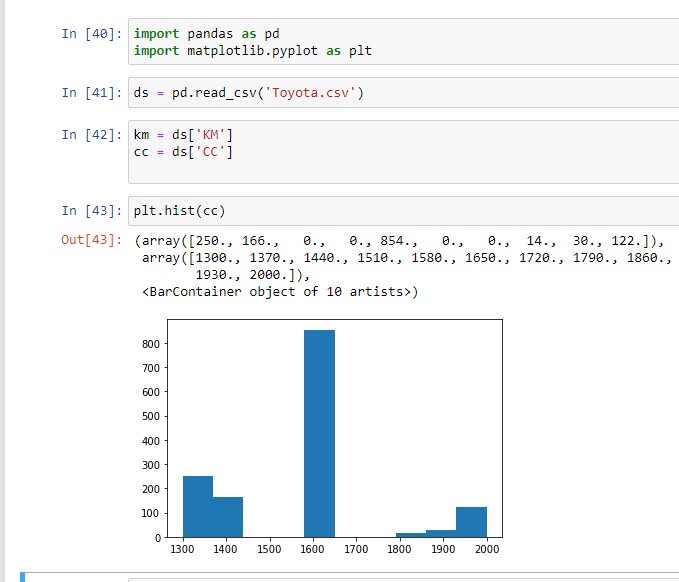
I used jupyter notebook to do all the tasks.

1. Scatter plot- Scatter plot of Price Vs Age

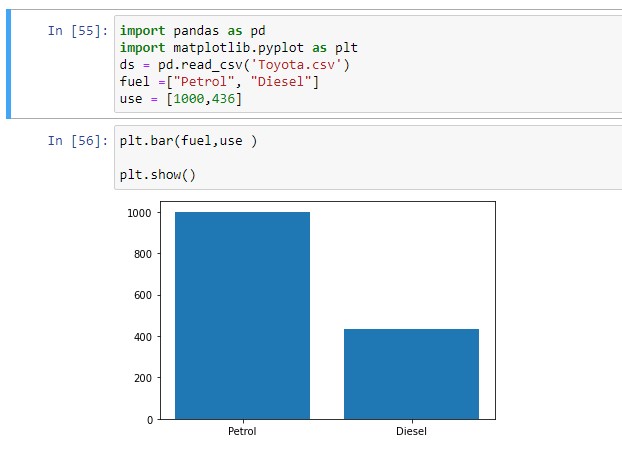


1. Histogram- for KM and CC





1. Bar plot- Bar plot for different fuel types



Q.5) Enlist some examples along with its purpose and properties (at least 10) of FOSS and proprietary software with respect to database.

**FOSS:**

Free and open-source software (FOSS) is a software that can be classified as both free software and open-source software.

Examples: Ubuntu, VLC Player, Android system, etc.

Advantages of FOSS:

1. Expansive licensing: Proprietary software licenses are usually quite restrictive in terms of use, number of users, type of machine and other. There is fee to own license of a proprietary software. Open-source software are free to own there is no restriction on how we use the software, we can install it on unlimited machines.
2. Transparency: Open-source development is carried out openly. As software code is openly available anyone who finds some bug can fix it for others. As development process is carried out publicly its development process is transparent. Users can easily communicate with product developers to understand their product decisions and offer opinions for betterment of software.
3. Source Code inspection: As source code is openly available anyone can view code of the software for better understanding of how the application works.
4. Source Code modification: We can also modify code after inspecting the code. One can modify code and for himself and also, he can add those changes to main version of software so that changes will be available for everyone for use.
5. Community: Foundation of open-source projects is community; it includes developers of software and also users. Users in community can easily share their feedback with developers so that developers can improve the software.
6. Redistribution rights: Open-source licenses allow users to perform some changes in software and again redistribute the software without taking permission of the original product owner.

**Proprietary Software:**

This type of software requires licenses for their use. Company or organization that owns the software provides rights to use the software to customer. Users can only install software only on limited number of machines and cannot redistribute it.

1. Increased Functionality and Convenience

Proprietary systems are easier to use and learn, leading to faster work processes. Skype, for example, is used by organizations worldwide. It takes minutes to sign up for an account and make international phone calls or conduct video interviews online. On top of that, your customers, suppliers and employees may already have a Skype account, so they know how to use it.

Open-source programs are trickier to use and may lack user-friendly features, affecting productivity in the workplace. Unskilled end users may find it difficult to navigate them and take full advantage of what they have to offer. After all, there is a reason so few people use Linux.

1. Superior Customer Support

Open-source software can be difficult to install and set up. Customizing it isn't easier either. Plus, your staff may not be familiar with the program and may need additional training.

The average employee lacks the expertise to use open-source programs. Therefore, your team members may need help with most tasks. They will spend hours trying to figure things out instead of focusing on the tasks at hand.

Proprietary software is more accessible and includes technical support. Most companies offering these programs provide dedicated sources, 24/7 assistance, live chat and user manuals. The antivirus program Bitdefender, for example, offers online resources, technical support around the clock and security-configuration services for enterprises. If your employees experience any issues, they can simply call or email the service provider.

1. Lower Maintenance Costs

As a small-business owner, you may prefer open-source software due to its low cost. Most programs are free or cost next to nothing. The downside is that you may end up paying a lot more for setup, maintenance and customization than you'd pay with proprietary software.

Some open-source programs are difficult to install and set up, so you may need to call an expert to do the job. In some cases, new hardware may be necessary to use the software. If your employees are not familiar with the program, they will need support and training, which may further increase the costs. Updating the software, testing new versions and applying patches isn't cheap either.

1. Stronger Competitive Advantage

Proprietary technology enables organizations to be more profitable, productive and innovative. This is particularly true for software-development companies, which often use custom programs at the core of their business model.

Even if you're not a software developer, you can still benefit from using proprietary systems. For example, you may hire a team to create software programs that integrate with your existing technology. This may improve work performance and productivity in your organization, streamline business processes and increase production. Furthermore, you may customize the program and add new features as your business changes.

1. Secure Financing for Your Business

Nearly one-third of start-ups close their doors because they run out of capital. Developing proprietary technology doesn't guarantee success, but it could make it easier to secure financing for your small business. Plus, you will be able to charge higher prices because no other company offers the same product as you do.

As it turns out, big data investors prefer to put their money in companies selling proprietary software — or at least something other than open-source software, such as proprietary add-ons. This kind of technology isn't restricted by what already exists in the market.