St. Francis Institute of Technology, Mumbai-400 103

**Department of Information Technology**

A.Y. 2021-2022

Class: TE-ITA/B, Semester: VI

Subject: **Data Science Lab**

**Experiment – 1: To implement Data Preparation using Numpy and Pandas.**

1. **Aim:** To implement Data Preparation using Numpy and Pandas.
2. **Objectives:** After study of this experiment, the student will be able to

* Understand Numpy concepts
* Understand Pandas concepts

1. **Outcomes:** After study of this experiment, the student will be able to

* Understand data preparation, Numpy and Pandas.

1. **Prerequisite:** Fundamentals of Python Programming and Database Management System.
2. **Requirements:** Python Installation,Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
3. **Pre-Experiment Exercise:**

**Brief Theory:**

Basic Concepts of Pandas and Numpy.

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**Software Installation:**

1. Python 3.6
   * This setup requires that your machine has python 3.6 installed on it. you can refer to this url <https://www.python.org/downloads/> to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly). To do that check this: <https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/>.
   * Setting up PATH variable is optional as you can also run program without it.
2. Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url <https://www.anaconda.com/download/>
3. You will also need to download and install below 2 packages after you install either python or anaconda from the steps above
   * Pandas
   * Numpy

* if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages

pip install pandas

pip install numpy

* if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages

conda install -c anaconda numpy

conda install -c anaconda pandas

1. Use Google Colab

**Dataset Used:**

**Iris Dataset:**

Iris Dataset is considered as the Hello World for data science. It contains five columns namely – Petal Length, Petal Width, Sepal Length, Sepal Width, and Species Type. Iris is a flowering plant, the researchers have measured various features of the different iris flowers and recorded them digitally.

**import** pandas **as** pd  
**import** numpy **as** np  
df = pd.read\_csv(**"https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv"**)  
  
df.info()  
  
**"""\*\*Data Inspection\*\*"""**df.head(5) *# head*df.shape  
  
df.columns  
  
df[**"sepal\_length"**].nunique()  
  
df[**"sepal\_length"**].unique()  
  
*# number of unique values alltogether*df.columns.nunique()  
  
*# value counts*df[**'species'**].value\_counts()  
  
**"""\*\*Dealing with NA values\*\*"""***# show null/NA values per column*df.isnull().sum()  
  
*# show NA values as % of total observations per column*df.isnull().sum()\*100/len(df)  
  
*# drop all rows containing null*df.dropna()  
  
*# drop all columns containing null*df.dropna(axis=1)  
  
*# drop columns with less than 5 NA values*df.dropna(axis=1, thresh=5)  
  
*# replace all na values with -9999*df.fillna(-9999)  
  
*# fill na values with NaN*df.fillna(np.NaN)  
  
*# fill na values with strings*df.fillna(**"data missing"**)  
  
*# fill missing values with mean column values*df.fillna(df.mean())  
  
**"""\*\*Column Operation\*\*"""***# select a column*df[**"sepal\_length"**]  
  
*# select multiple columns and create a new dataframe X*X = df[[**"sepal\_length"**, **"sepal\_width"**, **"species"**]]  
X  
  
*# select a column by column number*df.iloc[:, [1,3,4]]  
  
*# save all columns to a list*df.columns.tolist()  
  
*# sorting values by column "sepalW" in ascending order*df.sort\_values(by = **"sepal\_width"**, ascending = **True**)  
  
*# add new calculated column*df[**'newcol'**] = df[**"sepal\_length"**]\*2  
df  
  
*# create a conditional calculated column*df[**'newcol'**] = [**"short" if** i<3 **else "long" for** i **in** df[**"sepal\_width"**]]   
df  
  
**"""\*\*Row Operation (Sort, Filter, Slice)\*\*"""***# select rows 3 to 10*df.iloc[3:10,]  
  
*# select rows 3 to 49 and columns 1 to 3*df.iloc[3:50, 1:4]  
  
*# randomly select 10 rows*df.sample(10)  
  
*# find rows with specific strings*df[df[**"species"**].isin([**"setosa"**])]  
  
*# conditional filtering*df[df.sepal\_length >= 5]  
  
*# filtering rows with multiple values e.g. 0.2, 0.3*df[df[**"petal\_width"**].isin([0.2, 0.3])]  
  
*# multi-conditional filtering*df[(df.petal\_length > 1) & (df.species==**"setosa"**) | (df.sepal\_width < 3)]  
  
*# drop rows*df.drop(df.index[1]) *# 1 is row index to be deleted***"""\*\*Grouping\*\*"""***# data grouped by column "species"*X = df.groupby(**"species"**)  
X  
  
*# return mean values of a column ("sepal\_length" ) grouped by "species" column*df.groupby(**"species"**)[**"sepal\_length"**].mean()  
  
*# return mean values of ALL columns grouped by "species" category*df.groupby(**"species"**).mean()  
  
*# get counts in different categories*df.groupby(**"species"**).nunique()

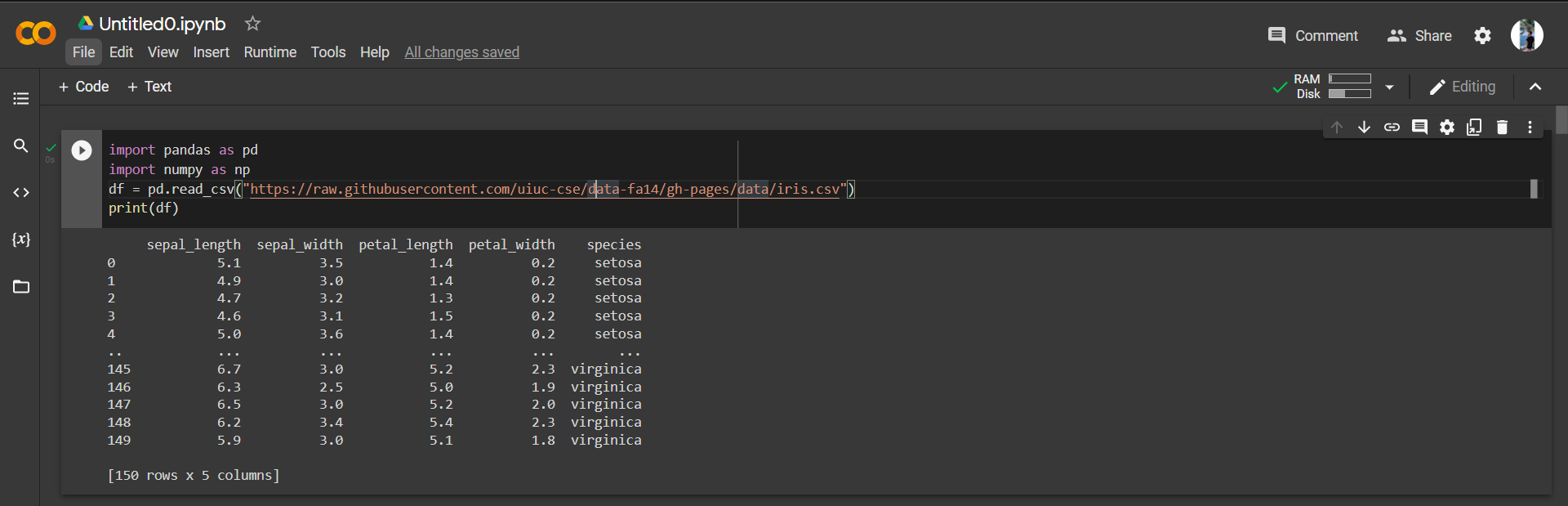
**Employee Dataset:**

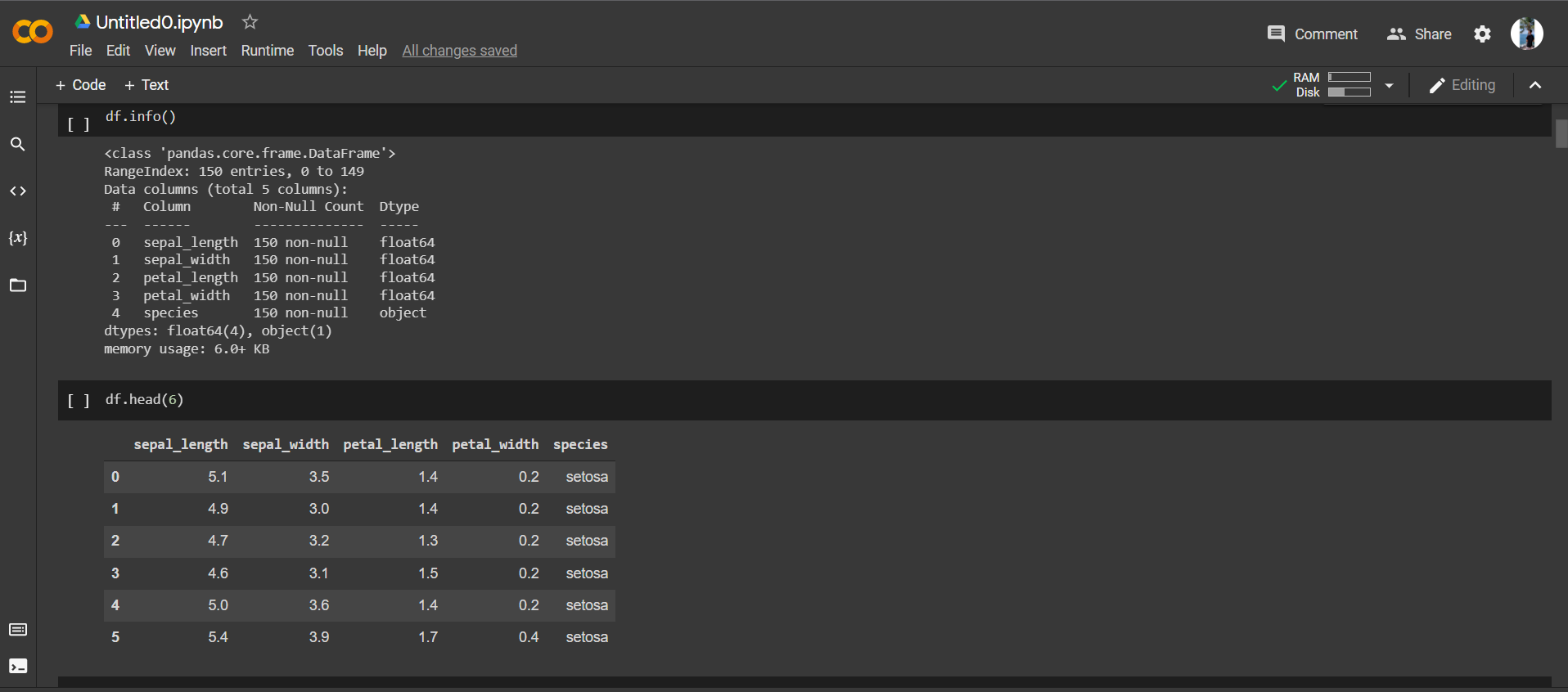
Employee dataset contains columns such as first name, gender, start date, last login, salary bonus, senior management and team. Some of the fields are null in the dataset.

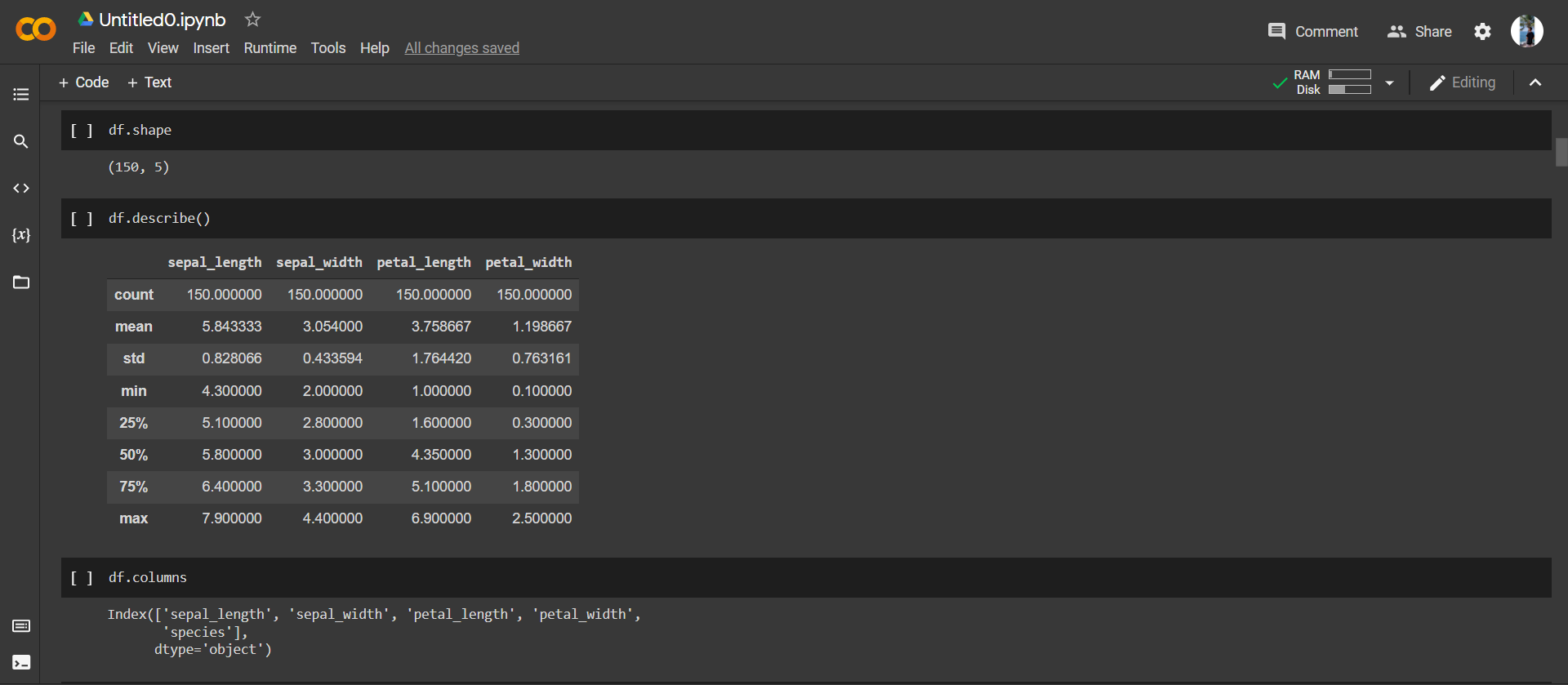
**import** pandas **as** pd  
df = pd.read\_csv(**"C:/Users/Vaishali/Desktop/AI-DS/employees.csv"**)  
print(df)  
  
print(df.describe())  
  
*#print(df.isNull())*print(pd.isnull(df[**'Team'**]))  
  
print(pd.notnull(df[**'Team'**]))  
  
print(df.fillna(1111))  
  
  
print(df.fillna(method=**'pad'**))  
  
*#import pandas as pd  
#df=pd.read\_csv("employees.csv")*print(df)  
  
df.fillna(method=**'bfill'**) *# check the output*print(df)  
  
df[**'Gender'**].fillna(**"No Gender"**,inplace=**True**)  
print(df)  
  
**import** numpy **as** np  
print(df.replace(to\_replace=np.NaN,value=**"SFIT"**))  
  
  
print(df)  
  
print(df.interpolate(method=**'linear'**,limit\_direction=**'forward'**))  
  
df1=pd.DataFrame({**"A"**:[12,23,**None**,5,6,**None**],  
 **"B"**:[34,**None**,2,34,5,67],  
 **"C"**:[67,54,33,**None**,77,98],  
 **"D"**:[45,87,65,33,23,**None**]  
  
})  
print(df1)  
  
print(df1.interpolate(method=**'linear'**,limit\_direction=**'forward'**))  
  
print(df1.dropna())

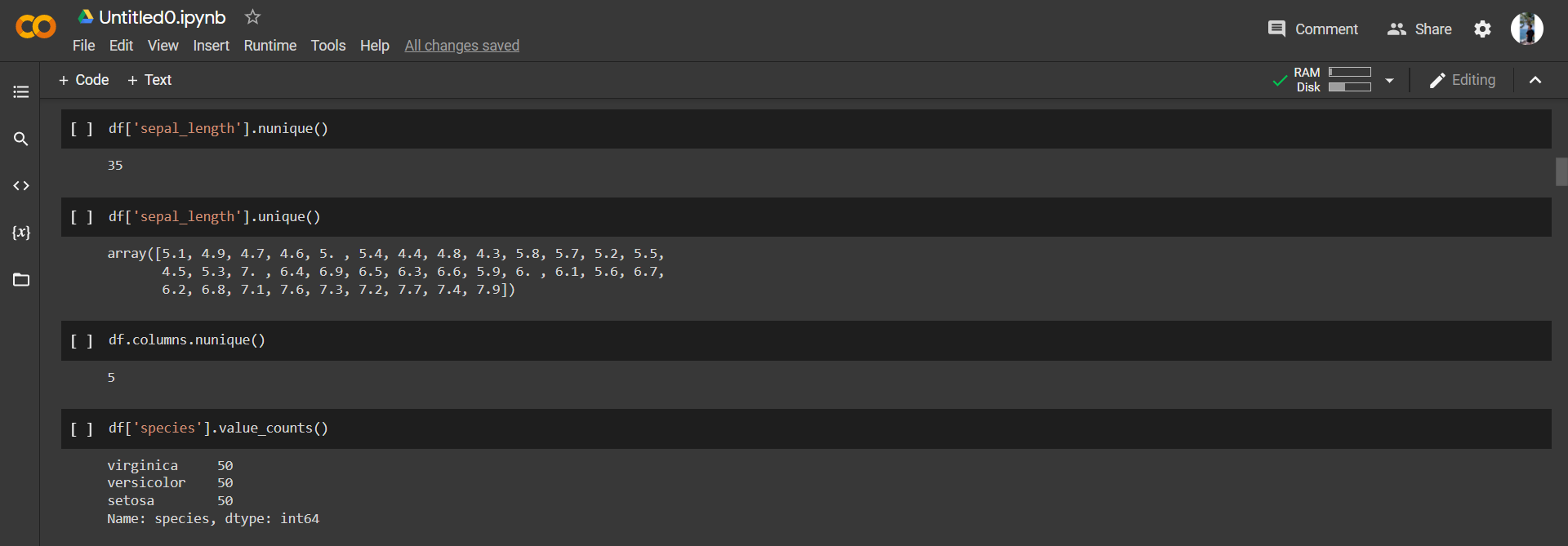
Paste Screenshots of above commands.

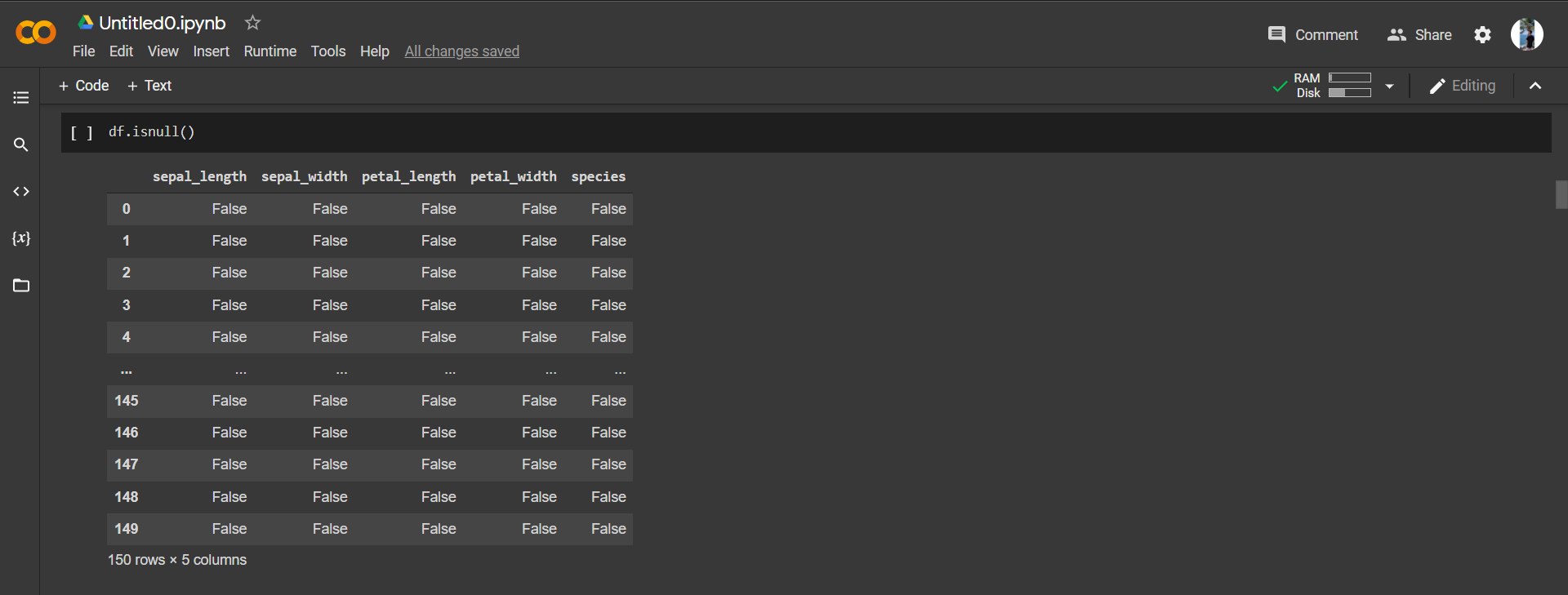
Iris Dataset:

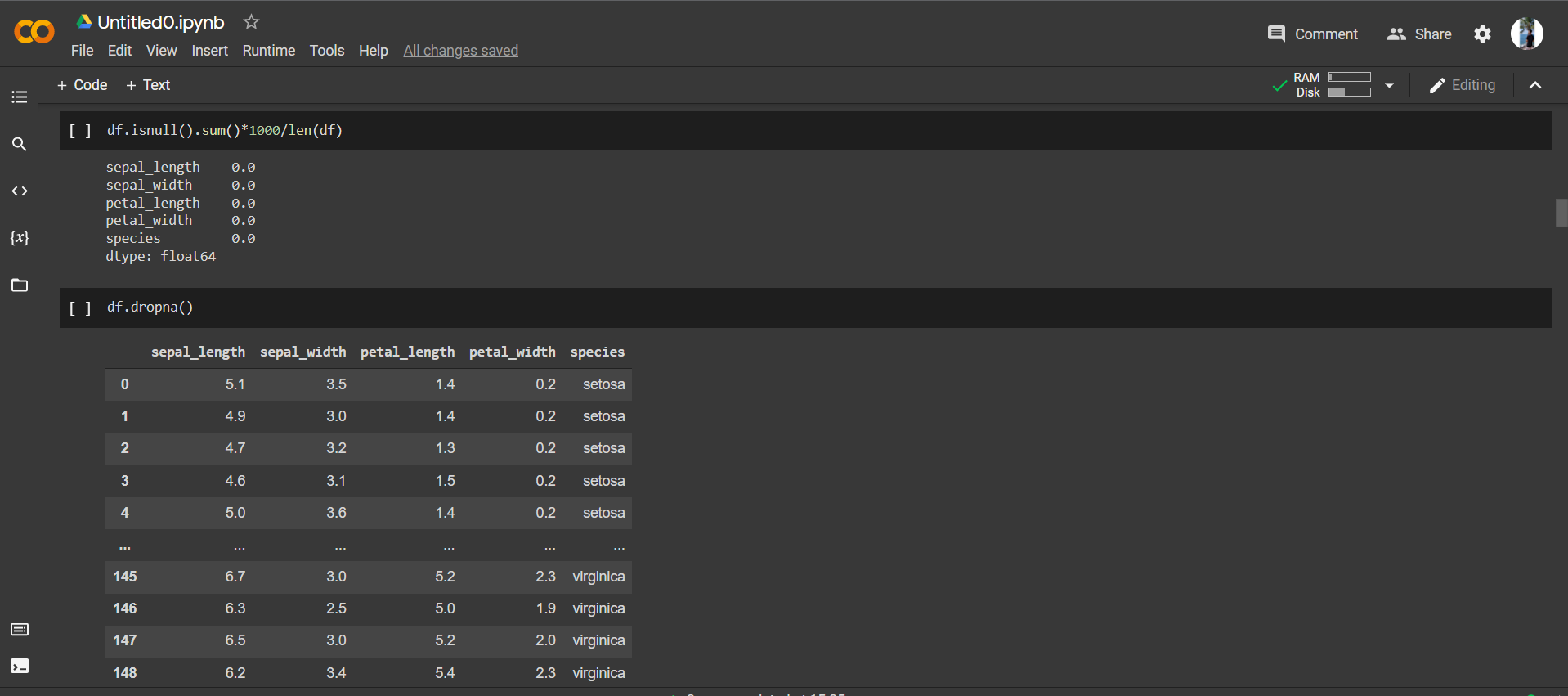


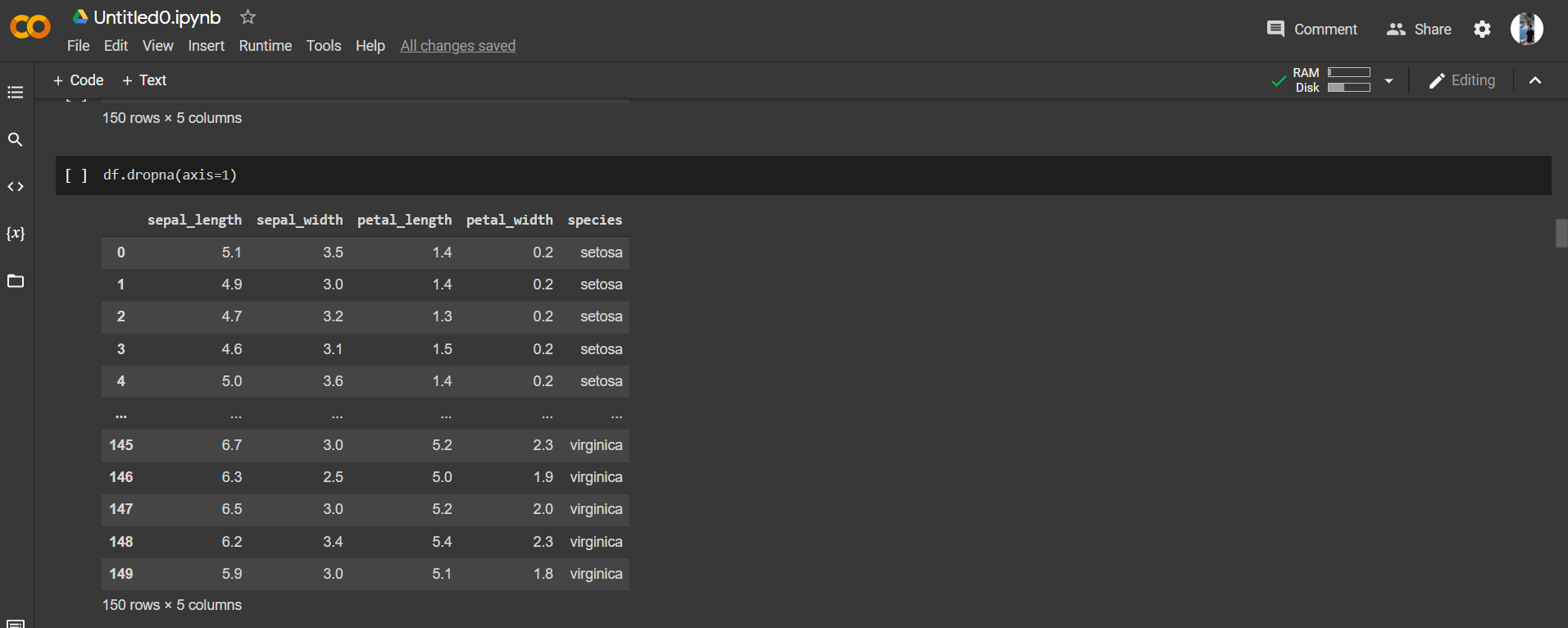


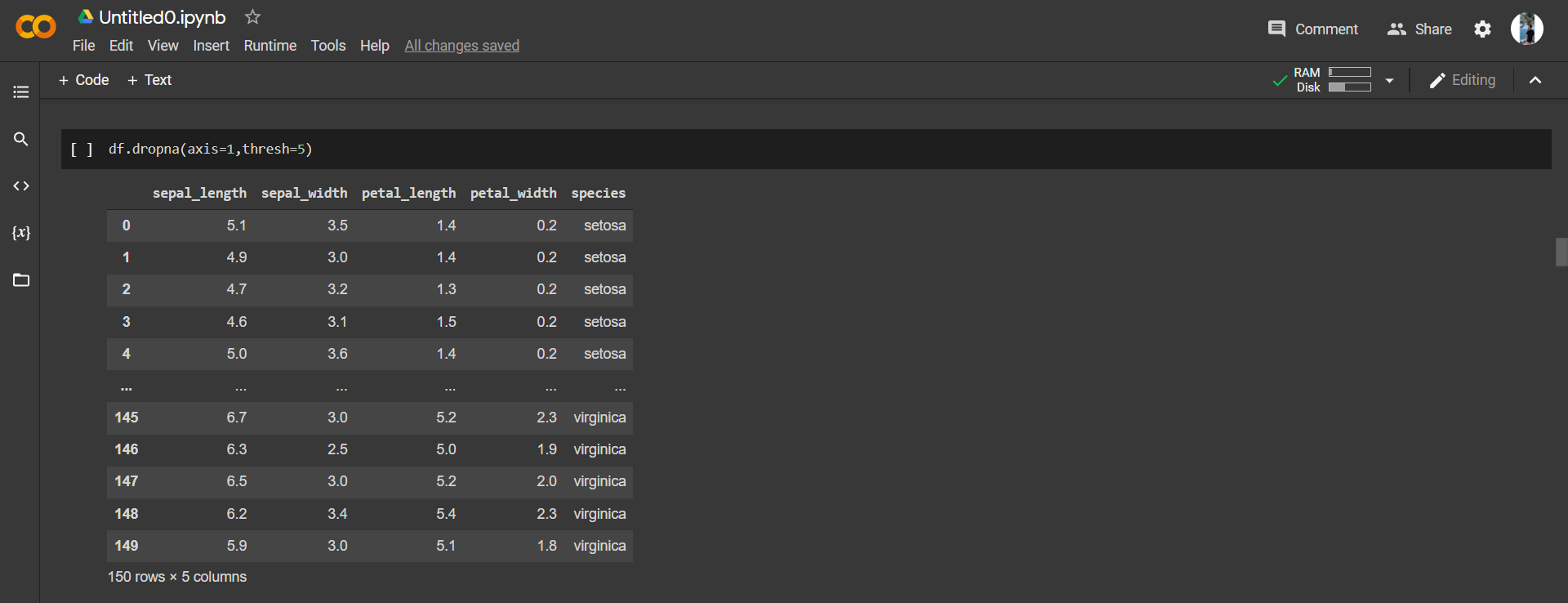


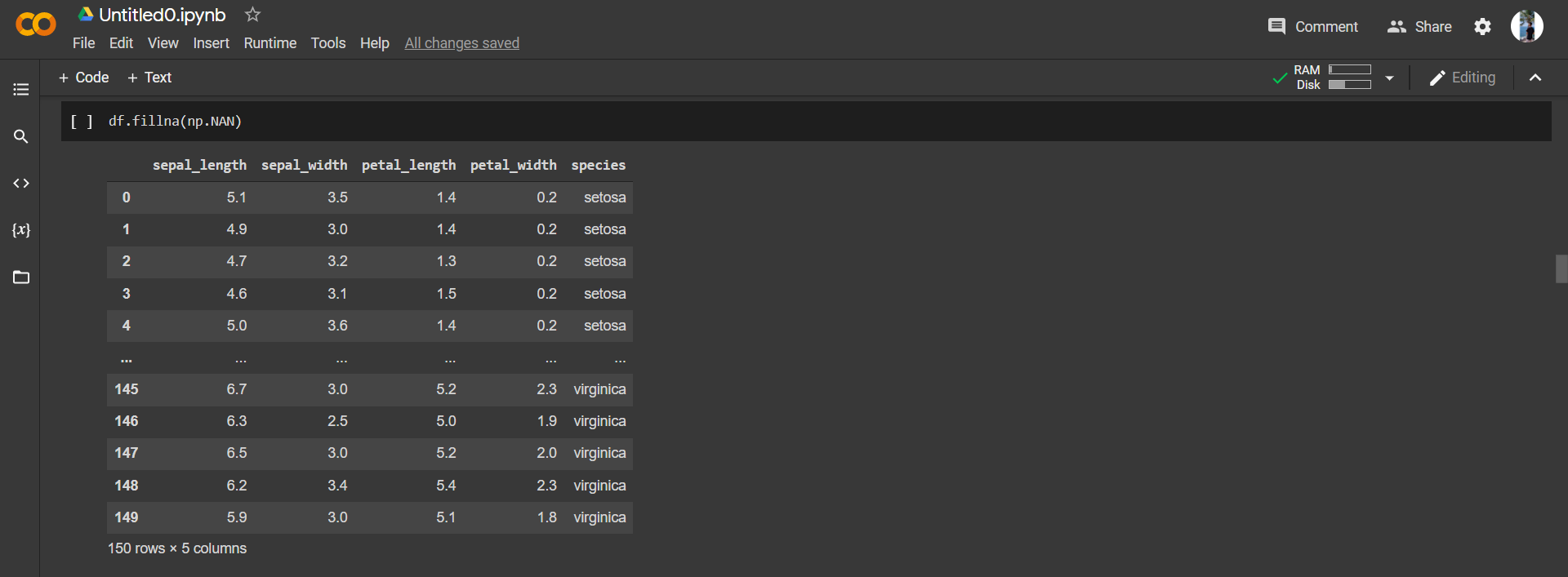


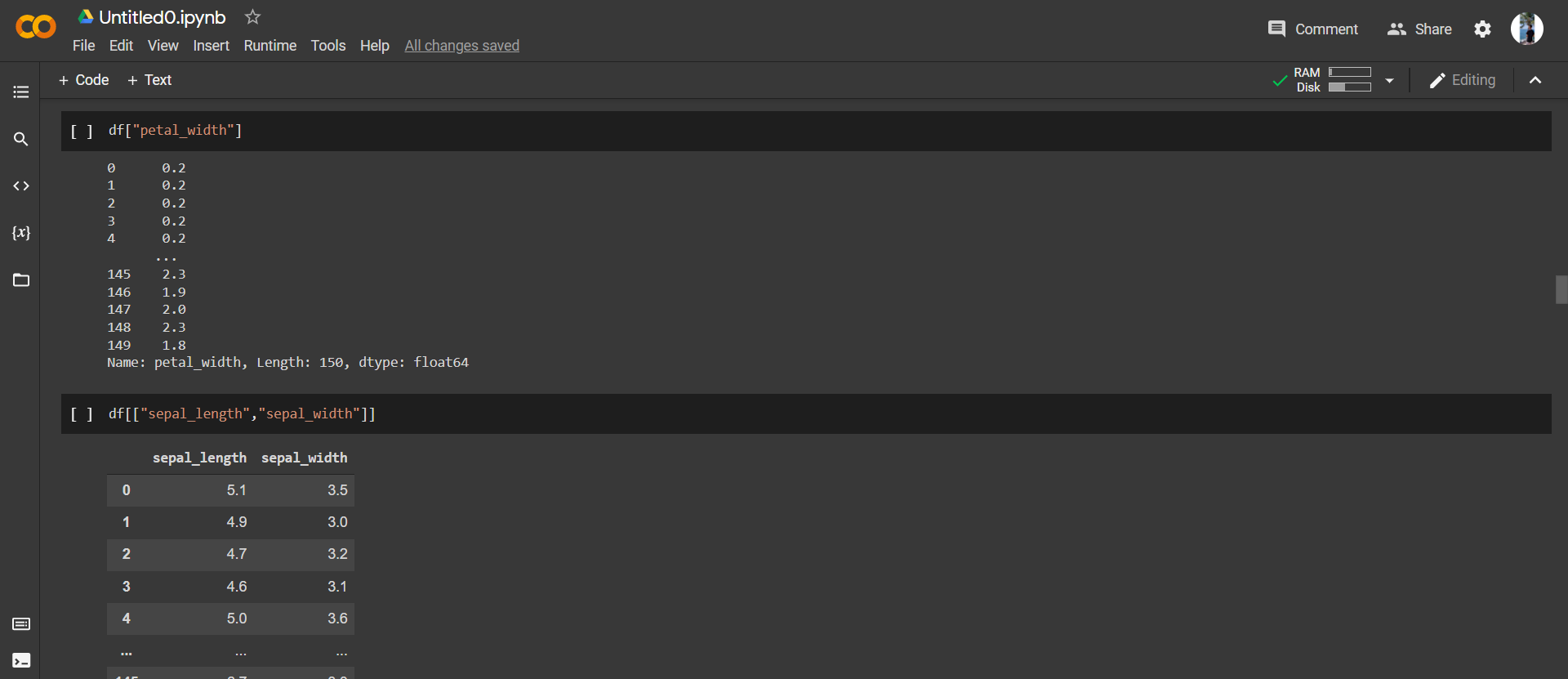


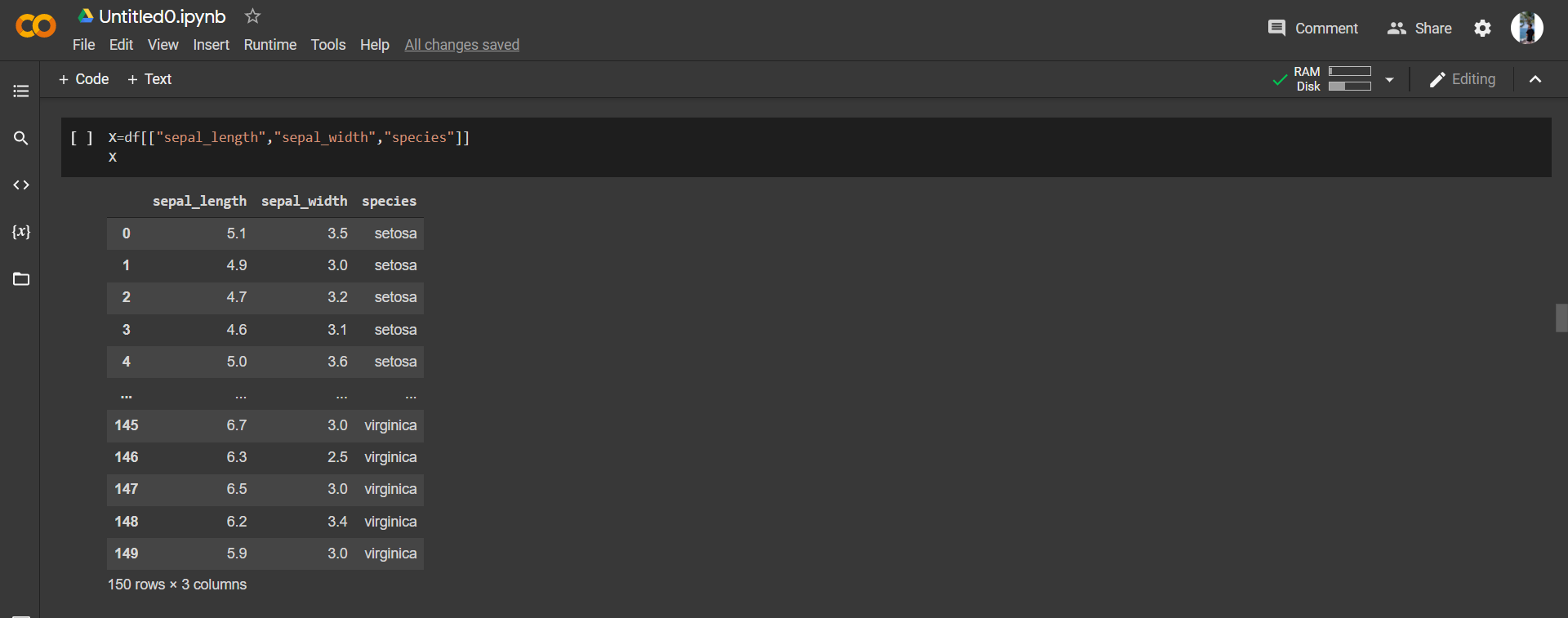


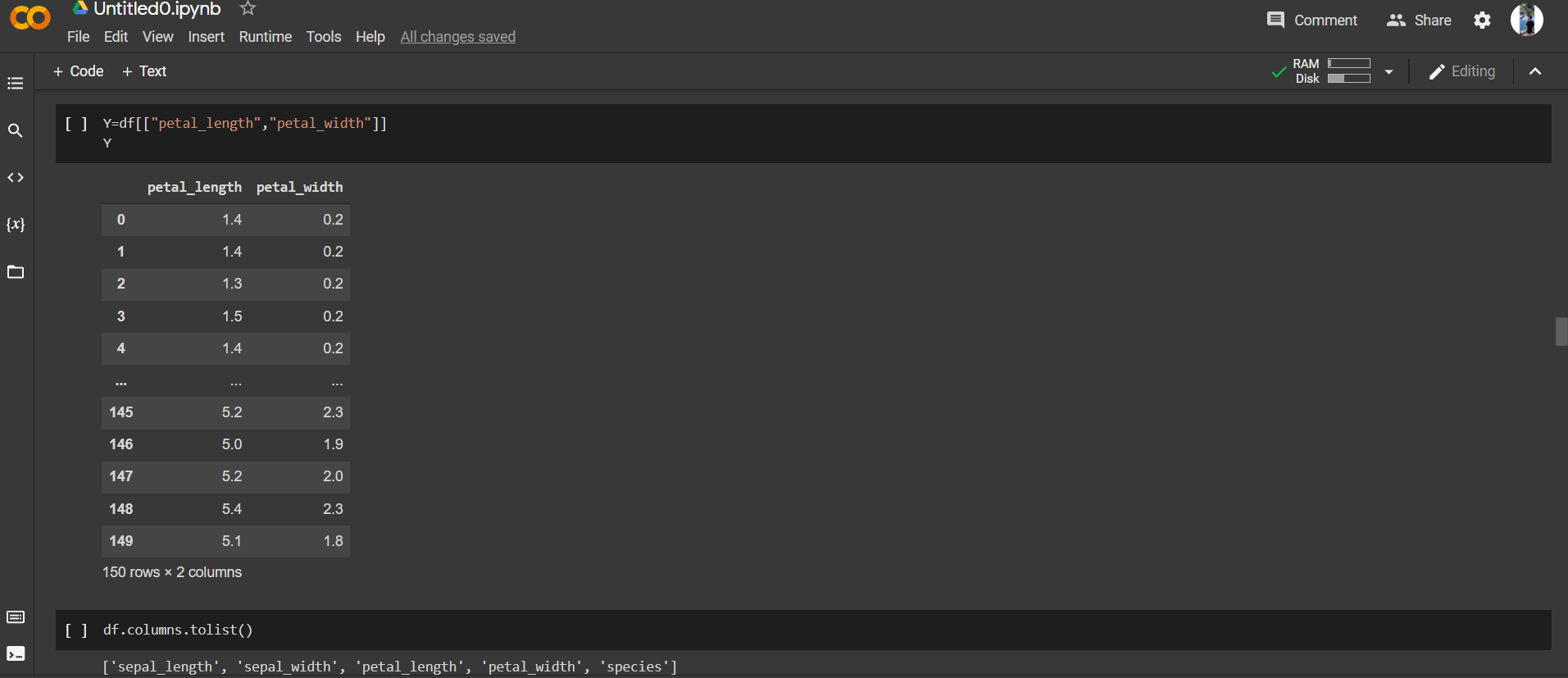


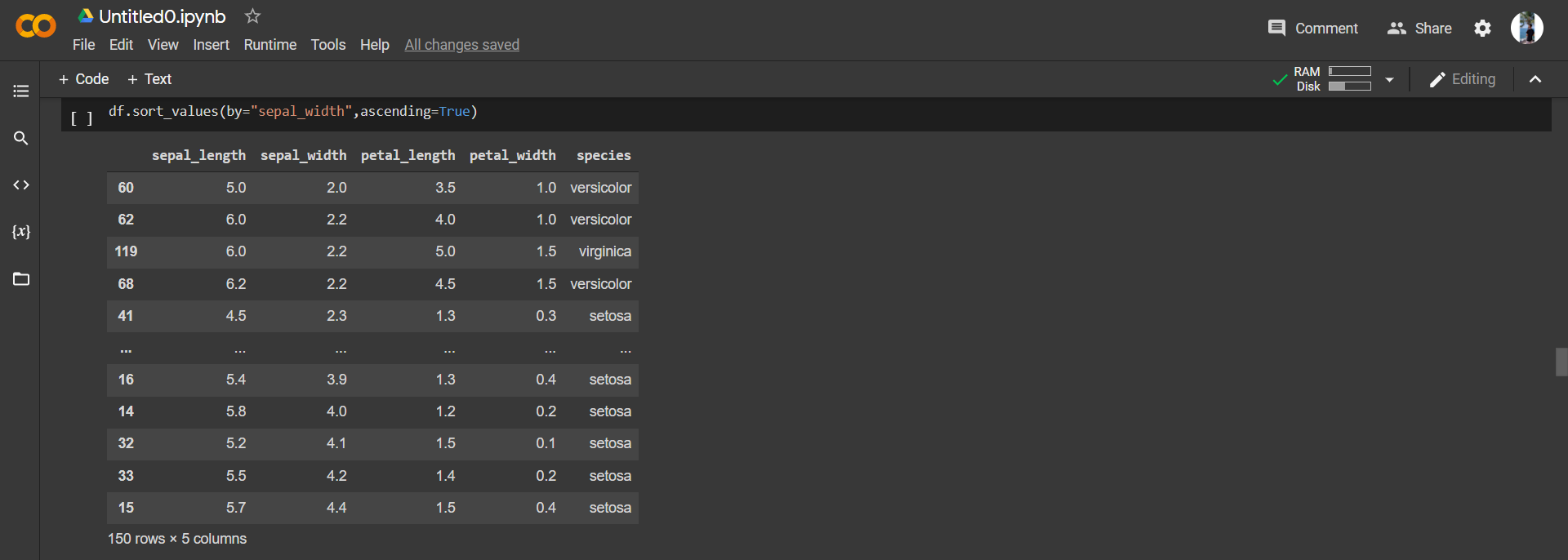


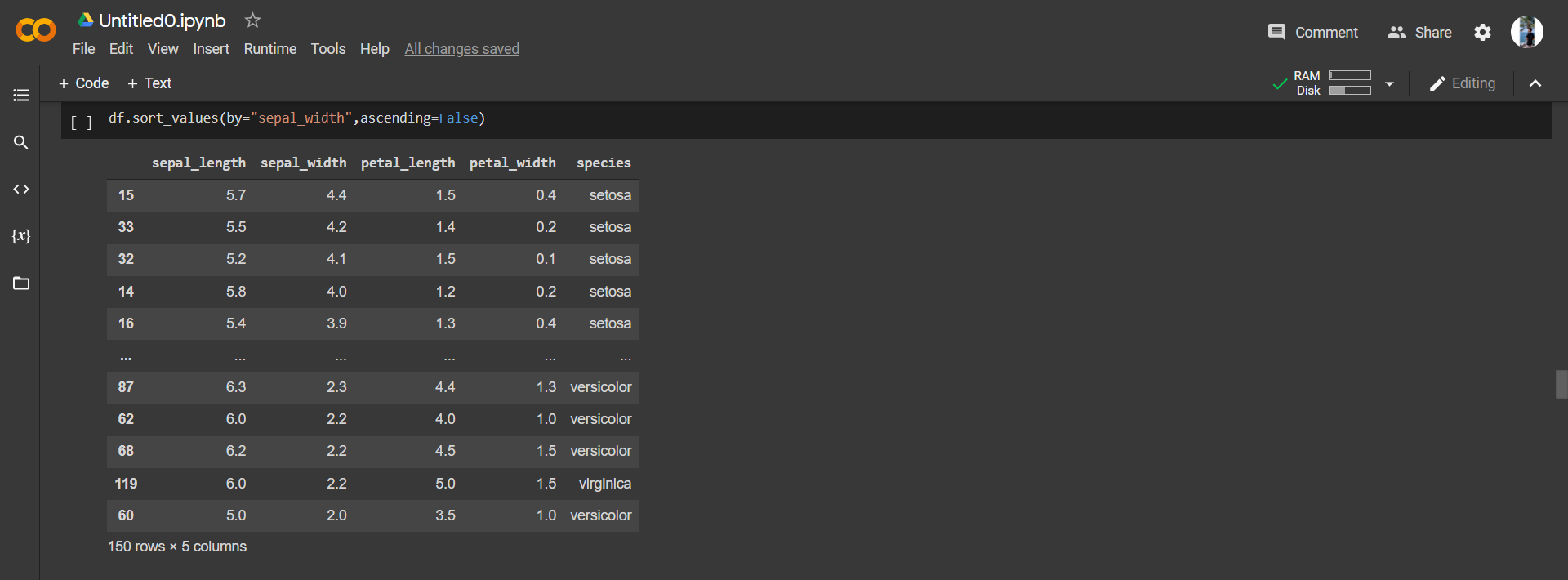


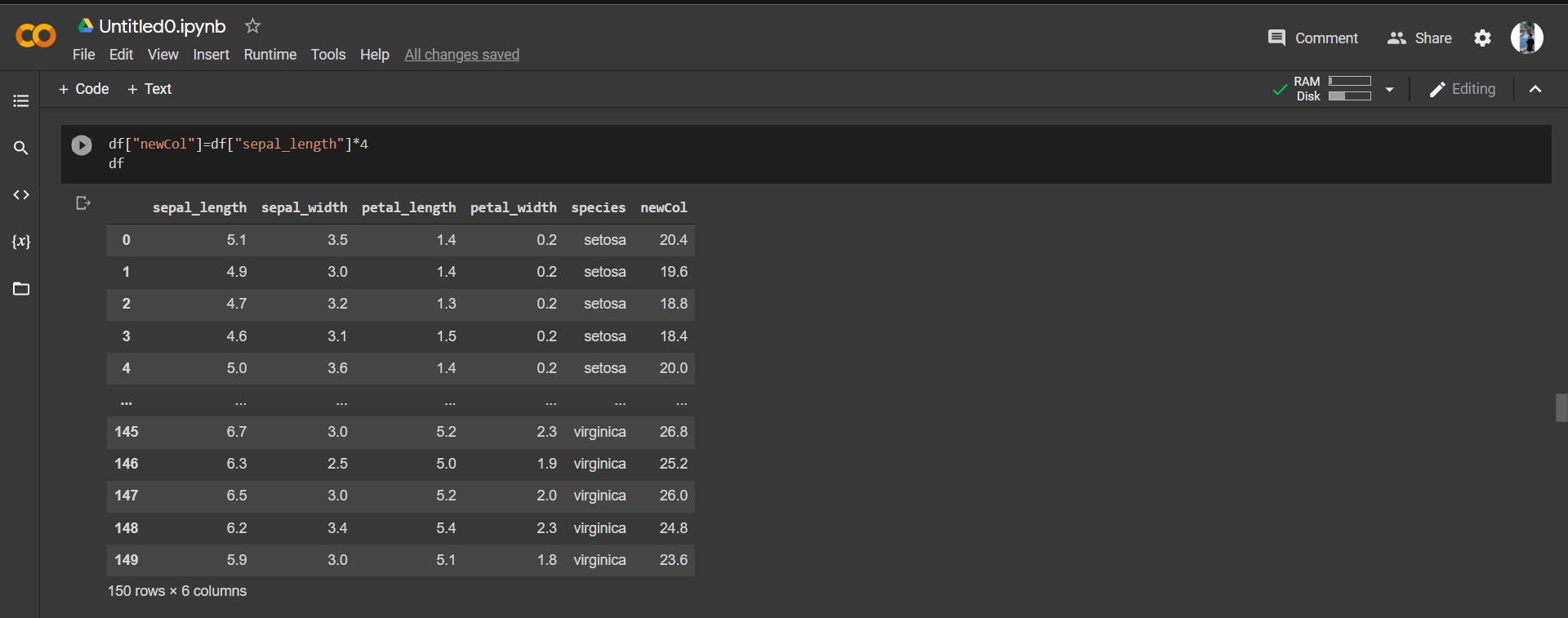


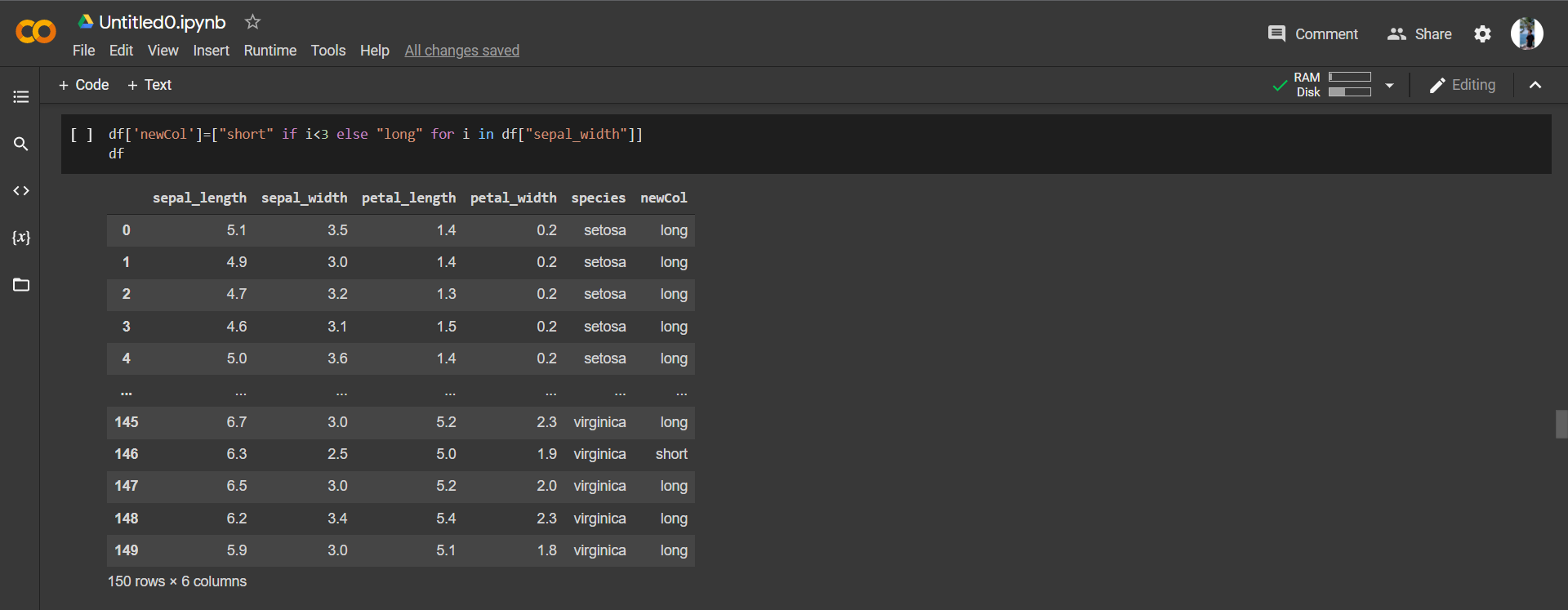


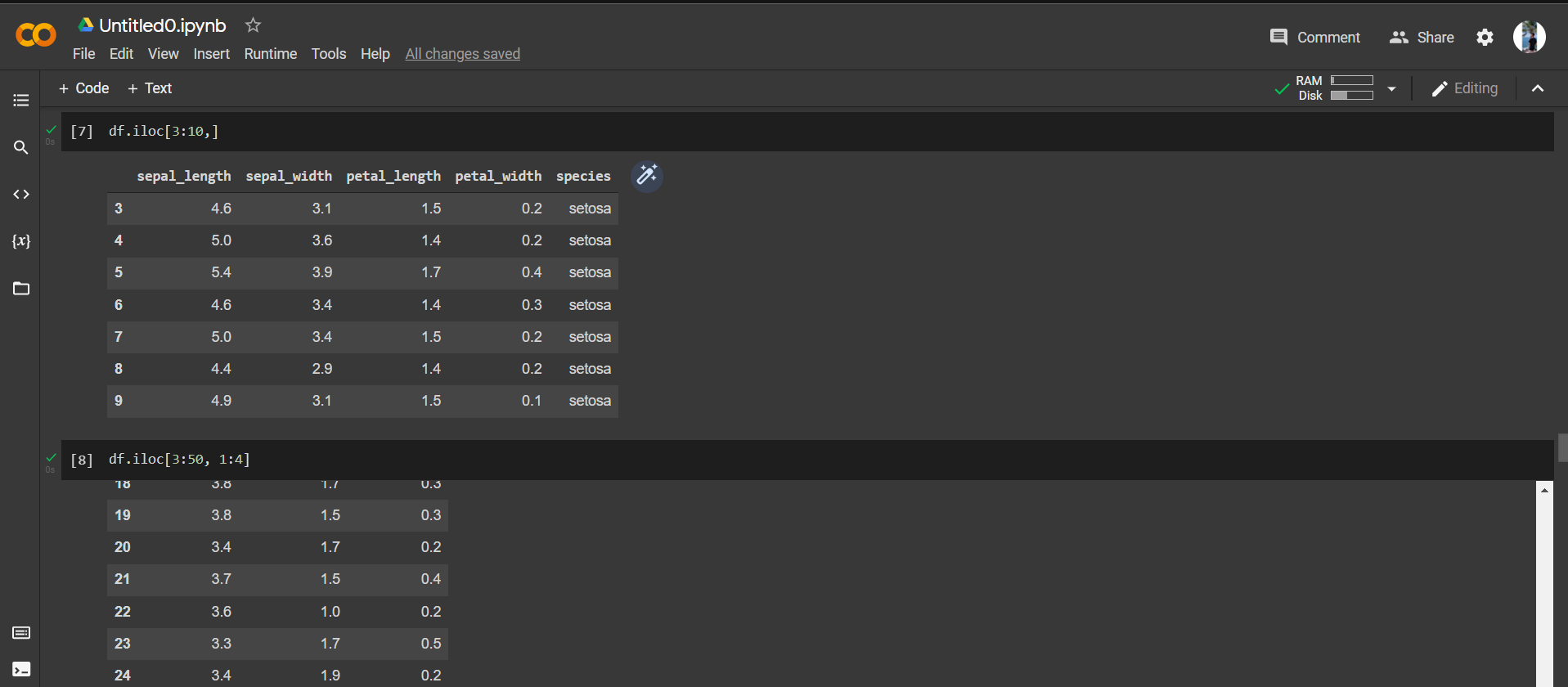


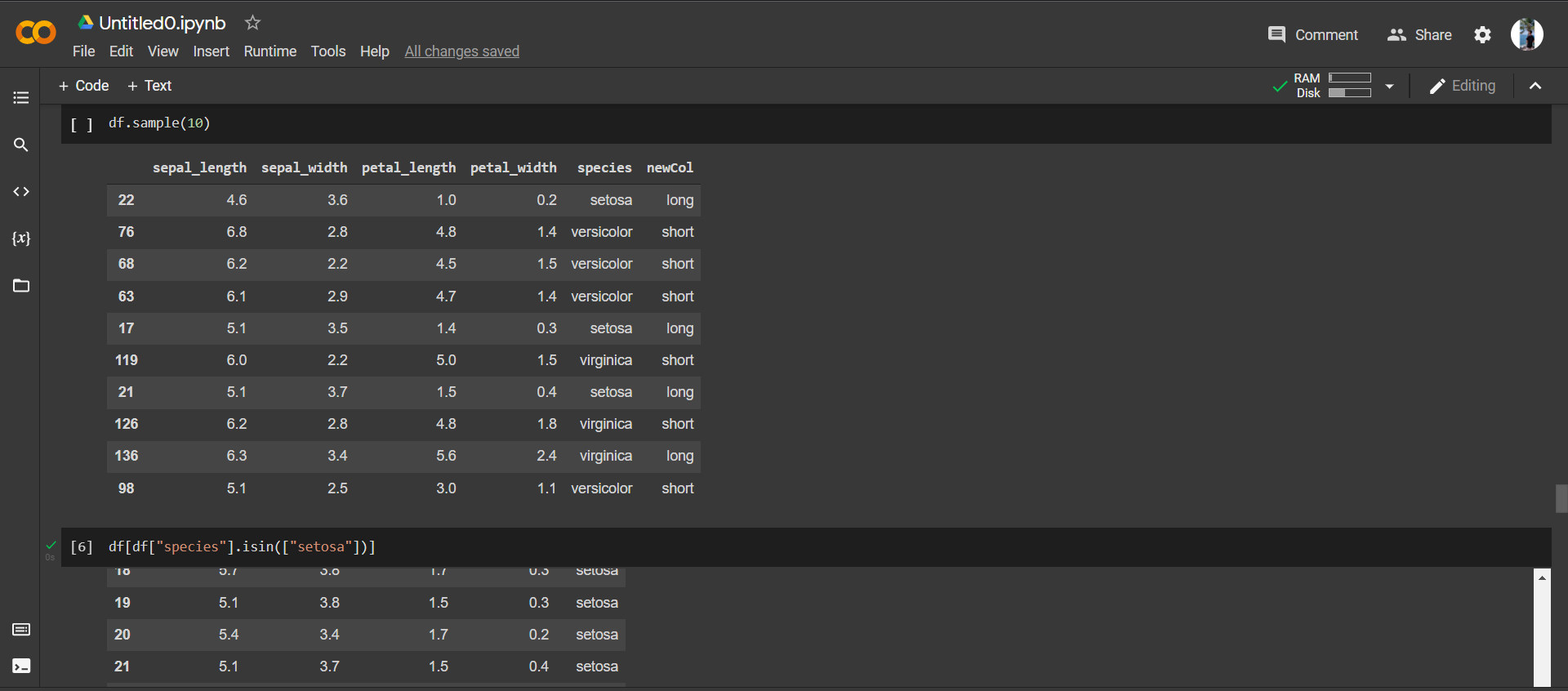


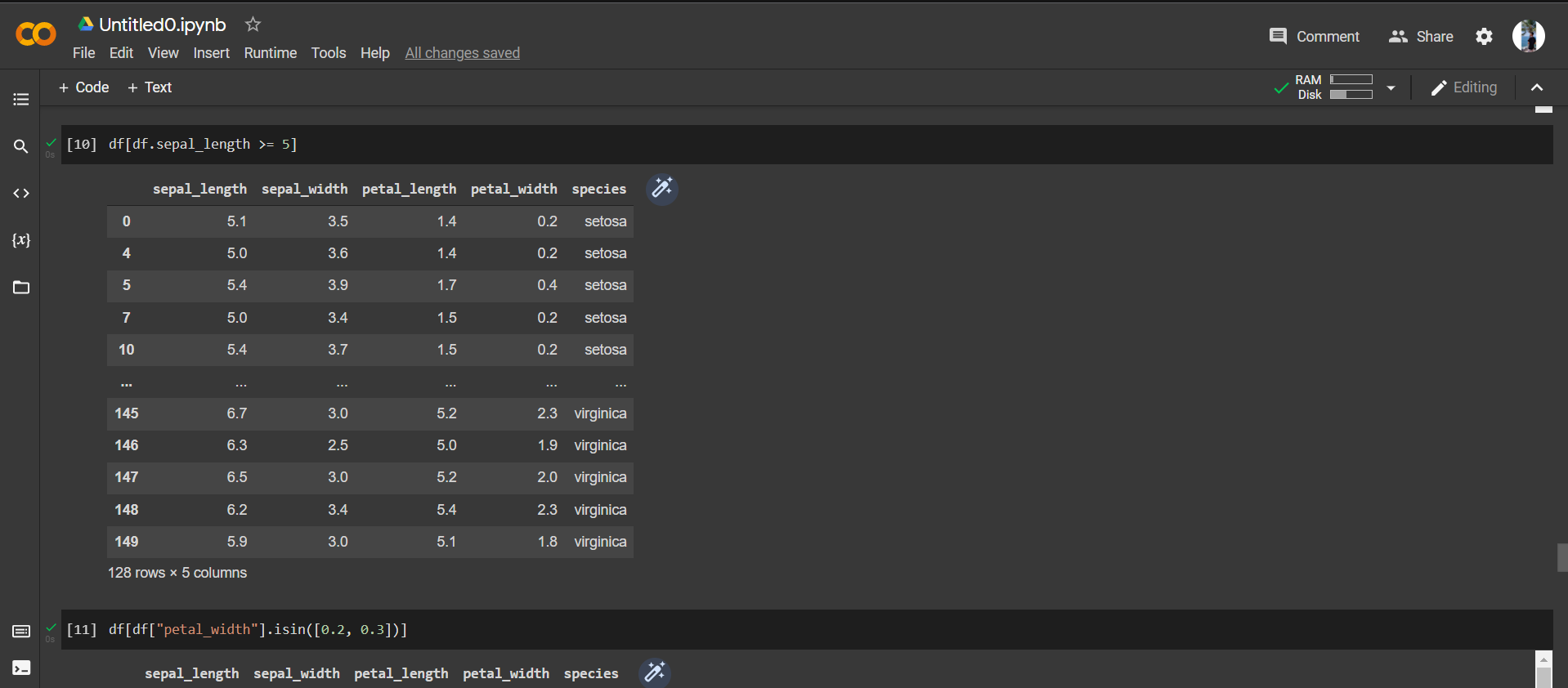


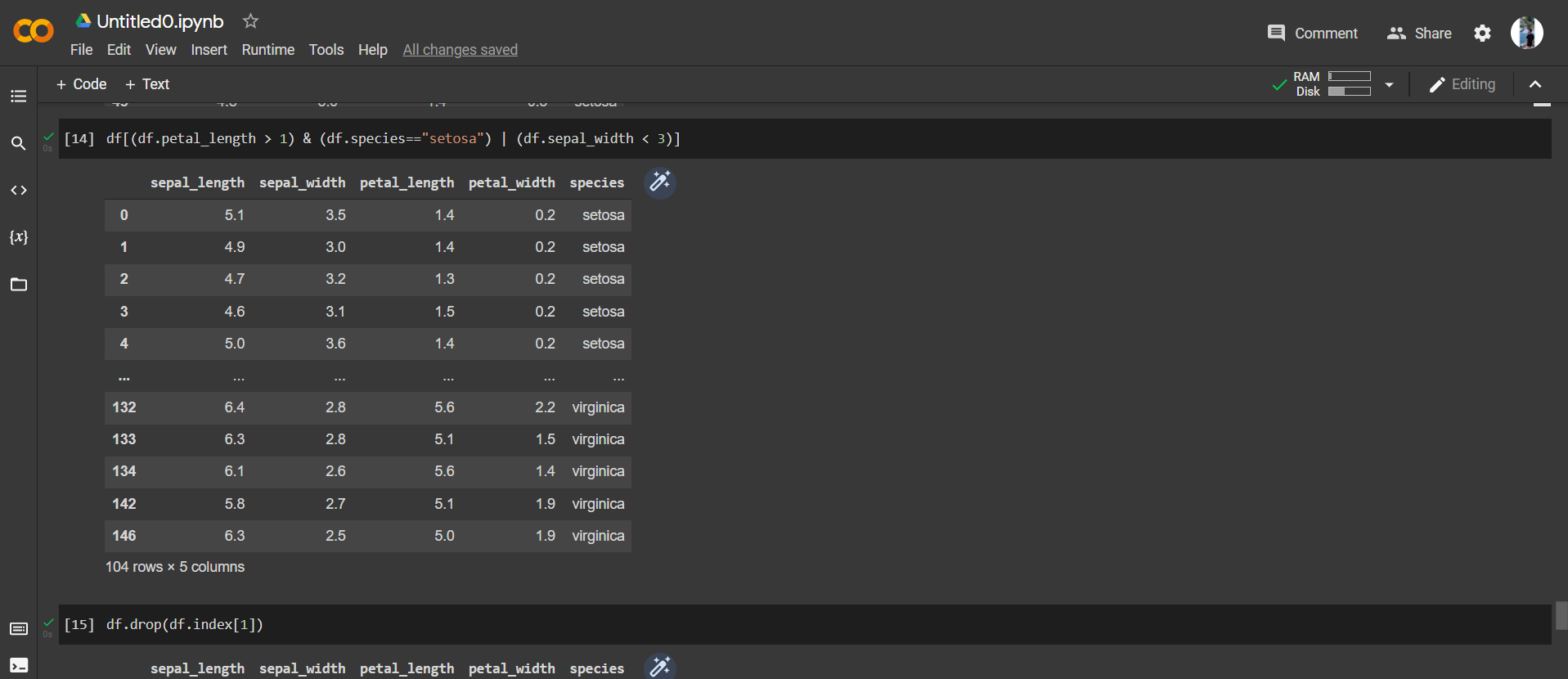


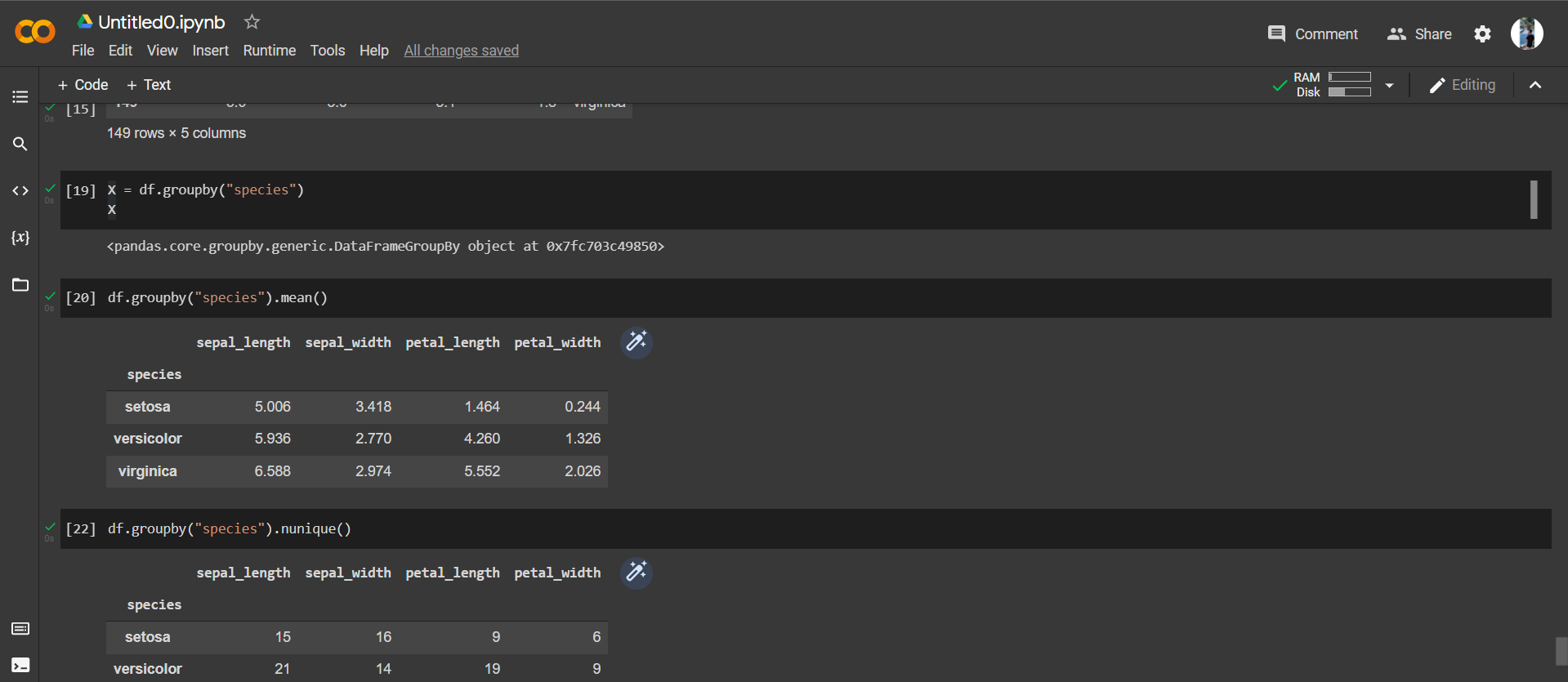




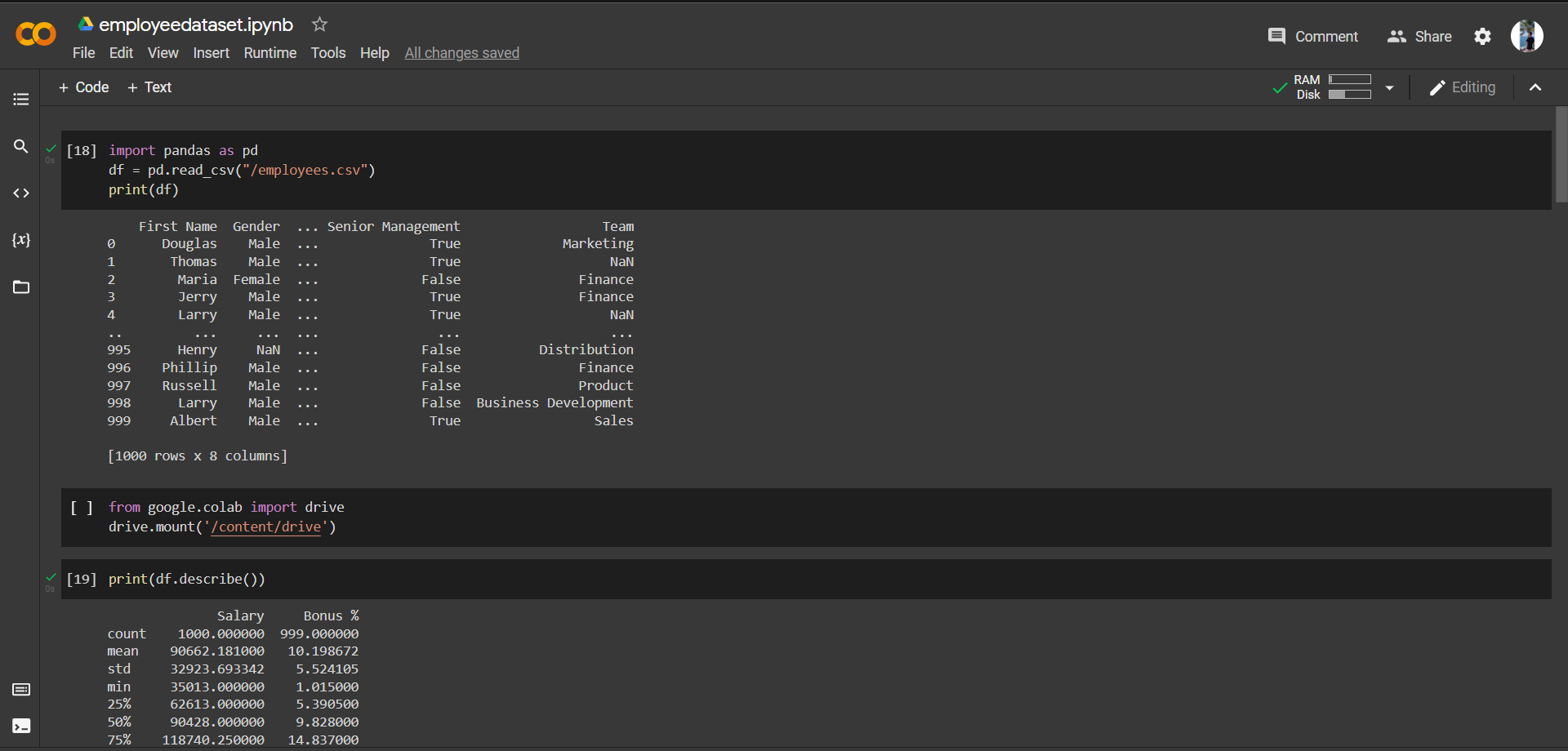


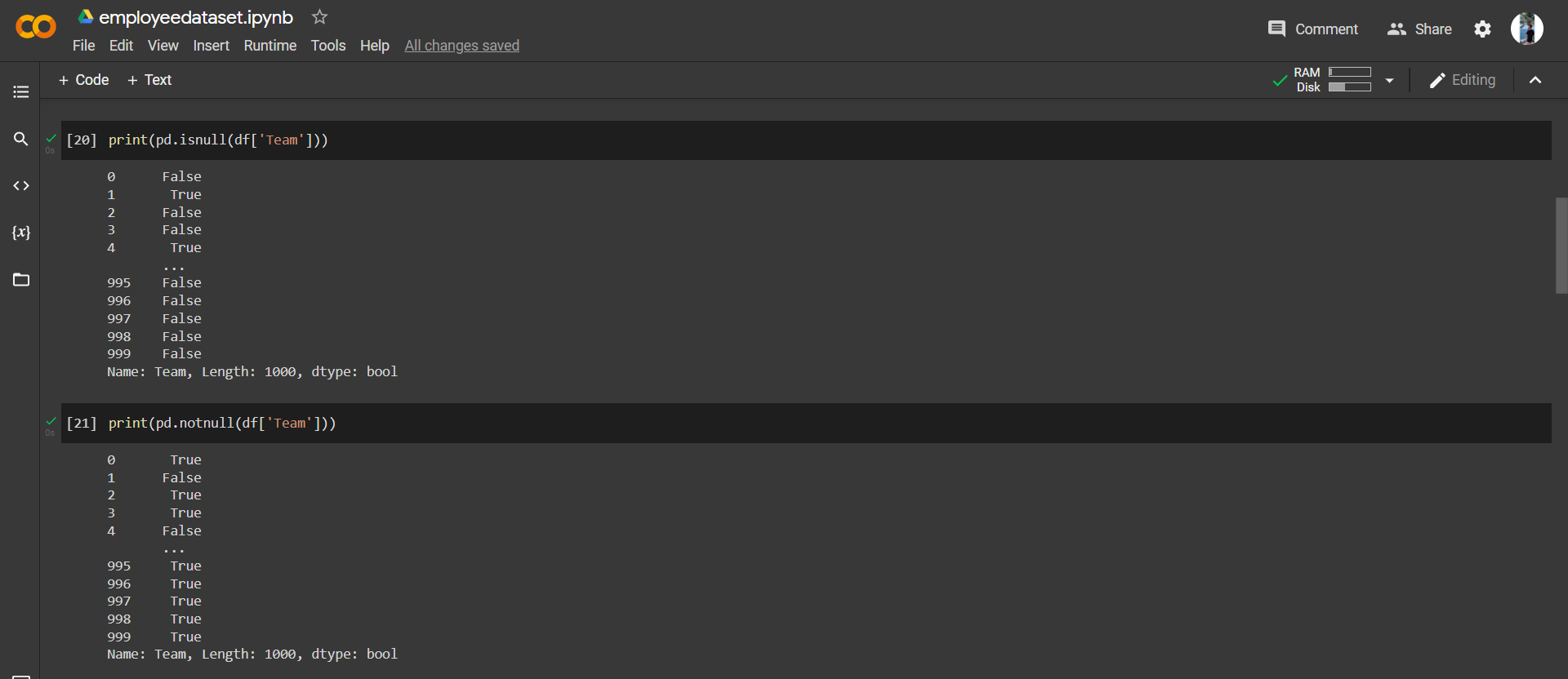


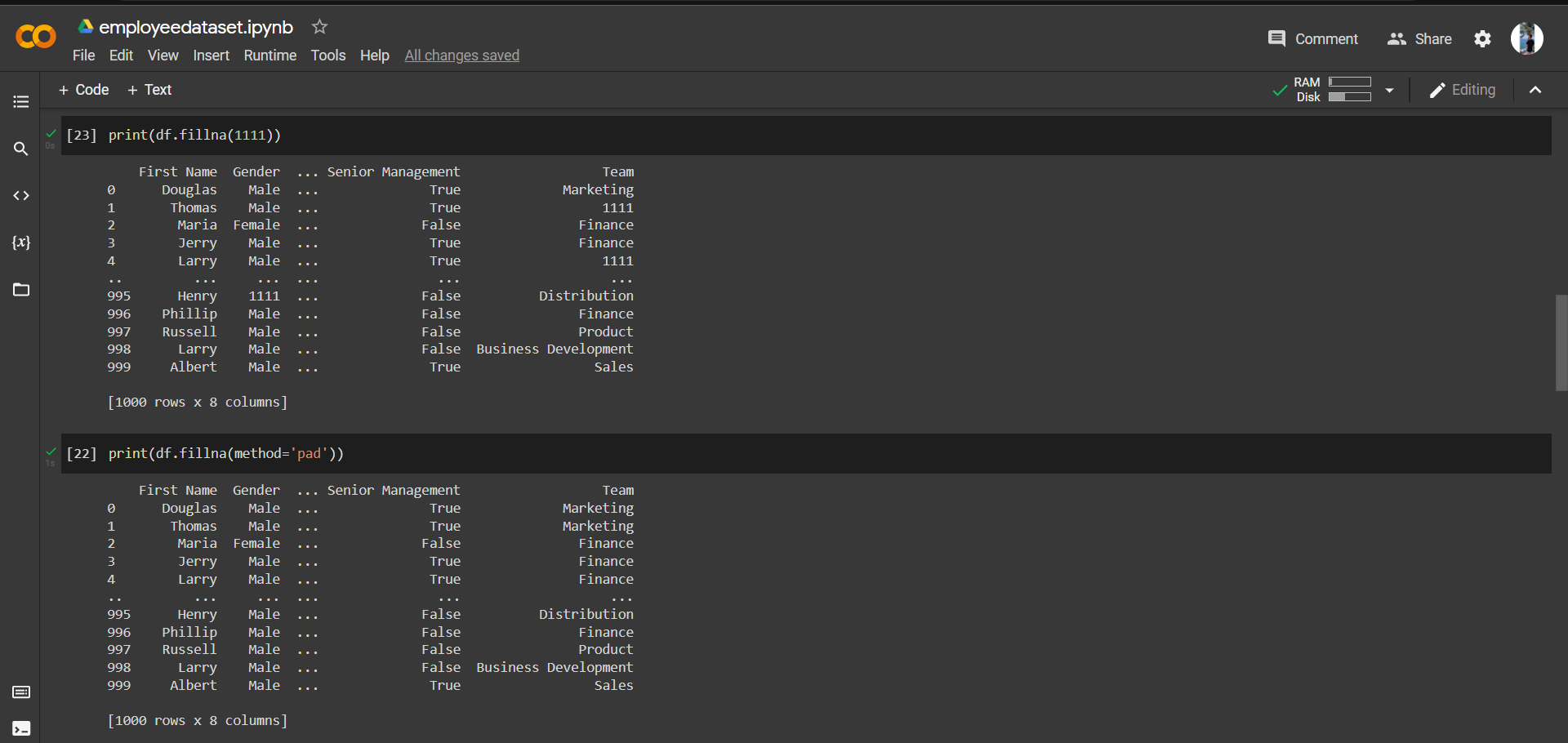


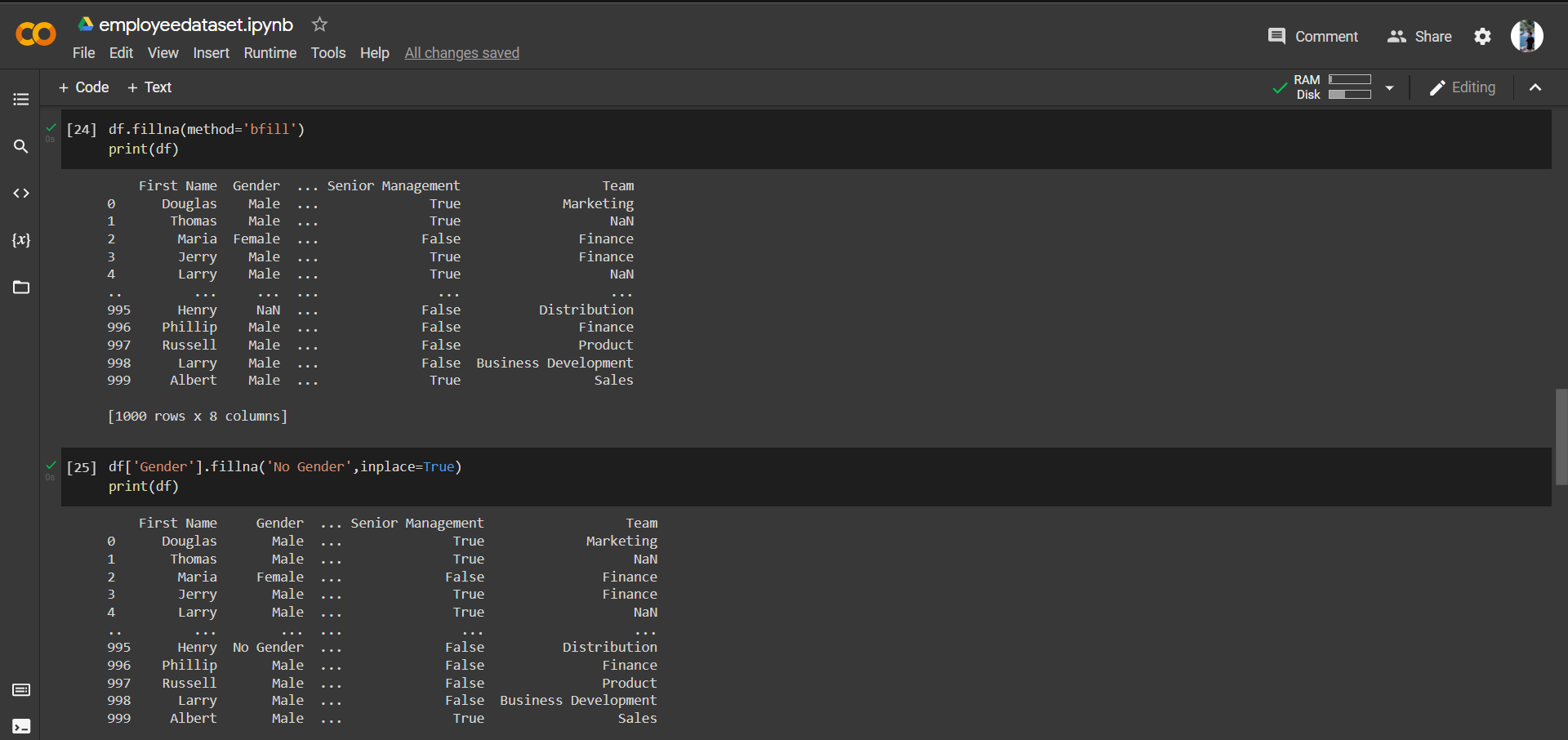


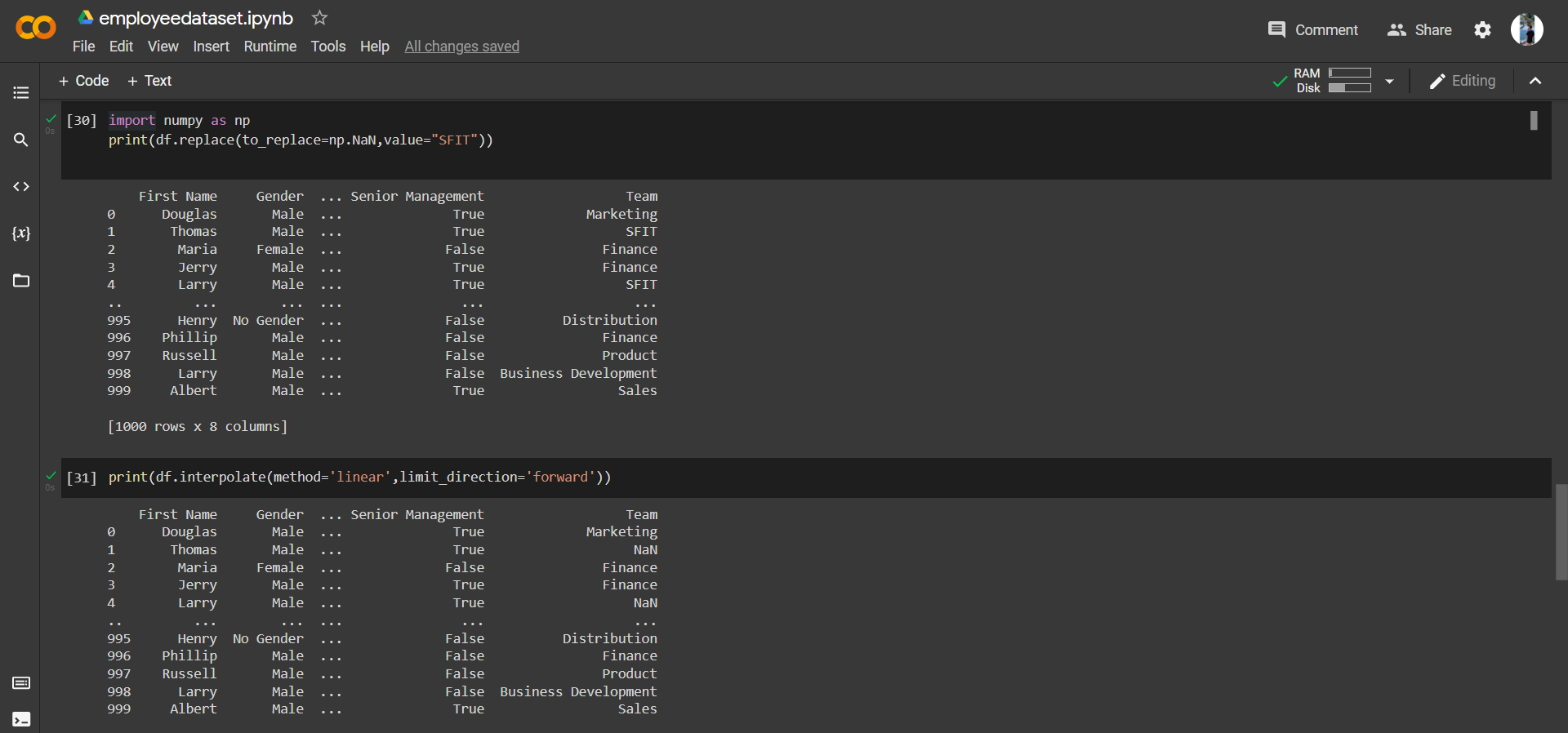
Employee

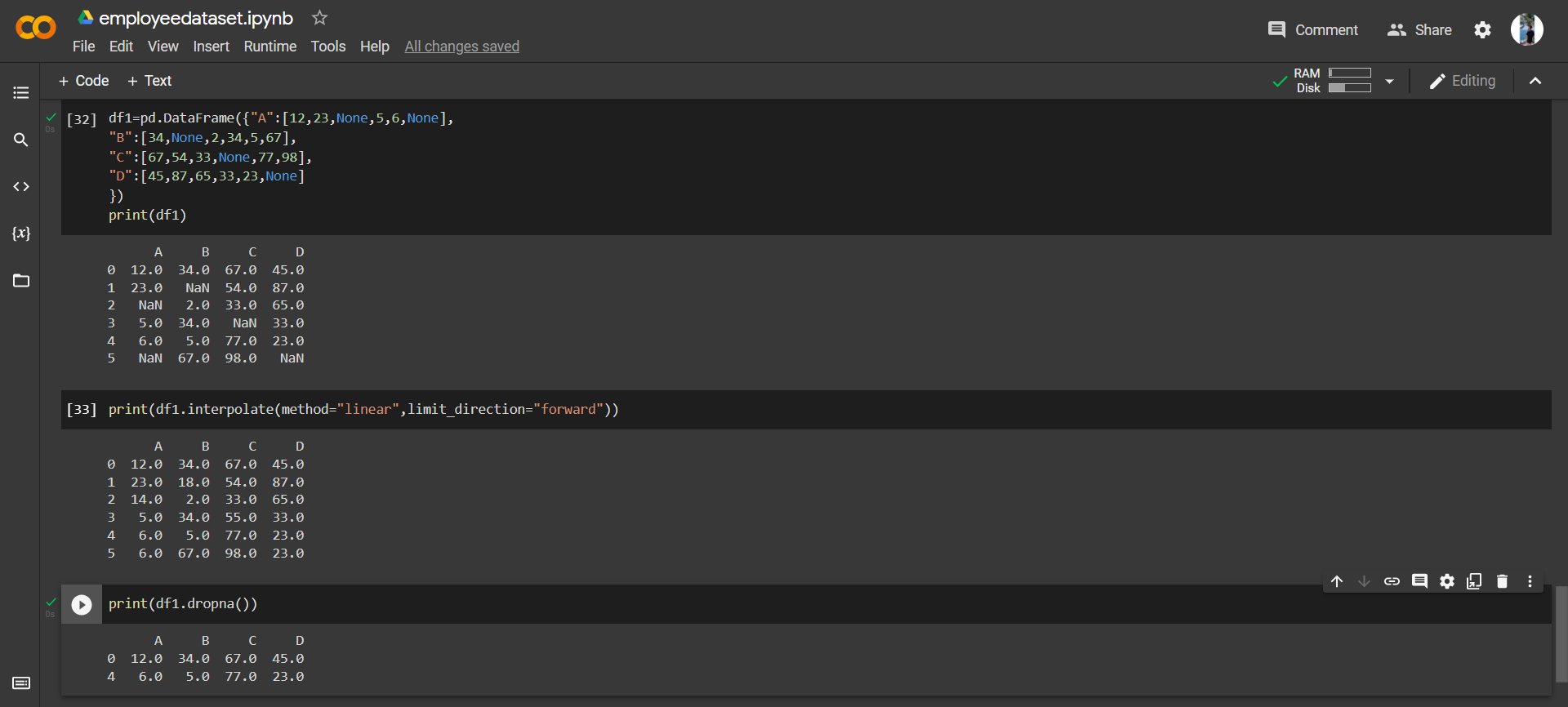












**8. Post-Experiments Exercise**

**A. Extended Theory: (Soft Copy)**

How to handle missing data in dataset? (Use Diabetes dataset & reference link)

Pandas treat None and NaN as essentially interchangeable for indicating missing or

null values. To facilitate this convention, there are several useful functions for detecting,

removing, and replacing null values in Pandas DataFrame :

● isnull()

● notnull()

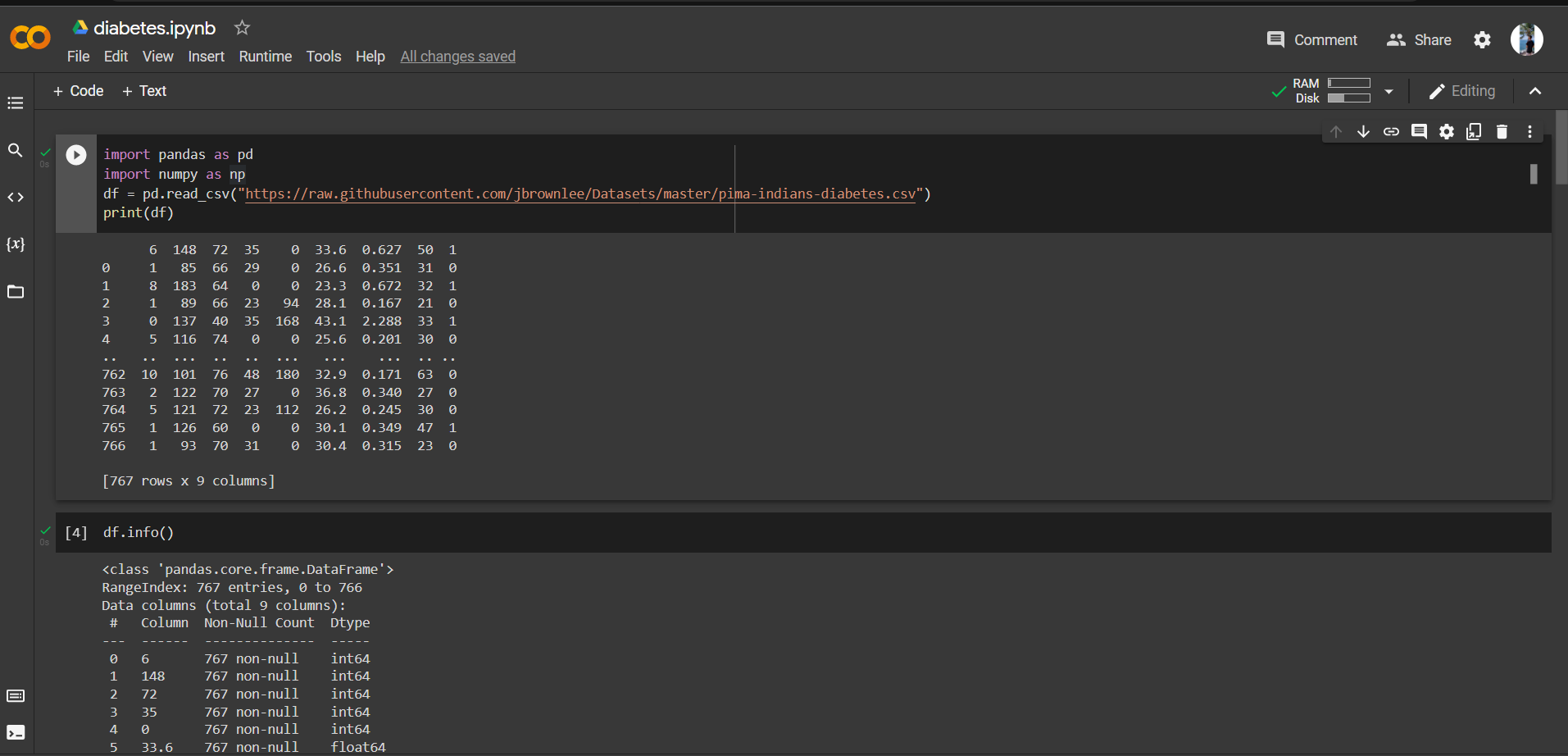
● dropna()

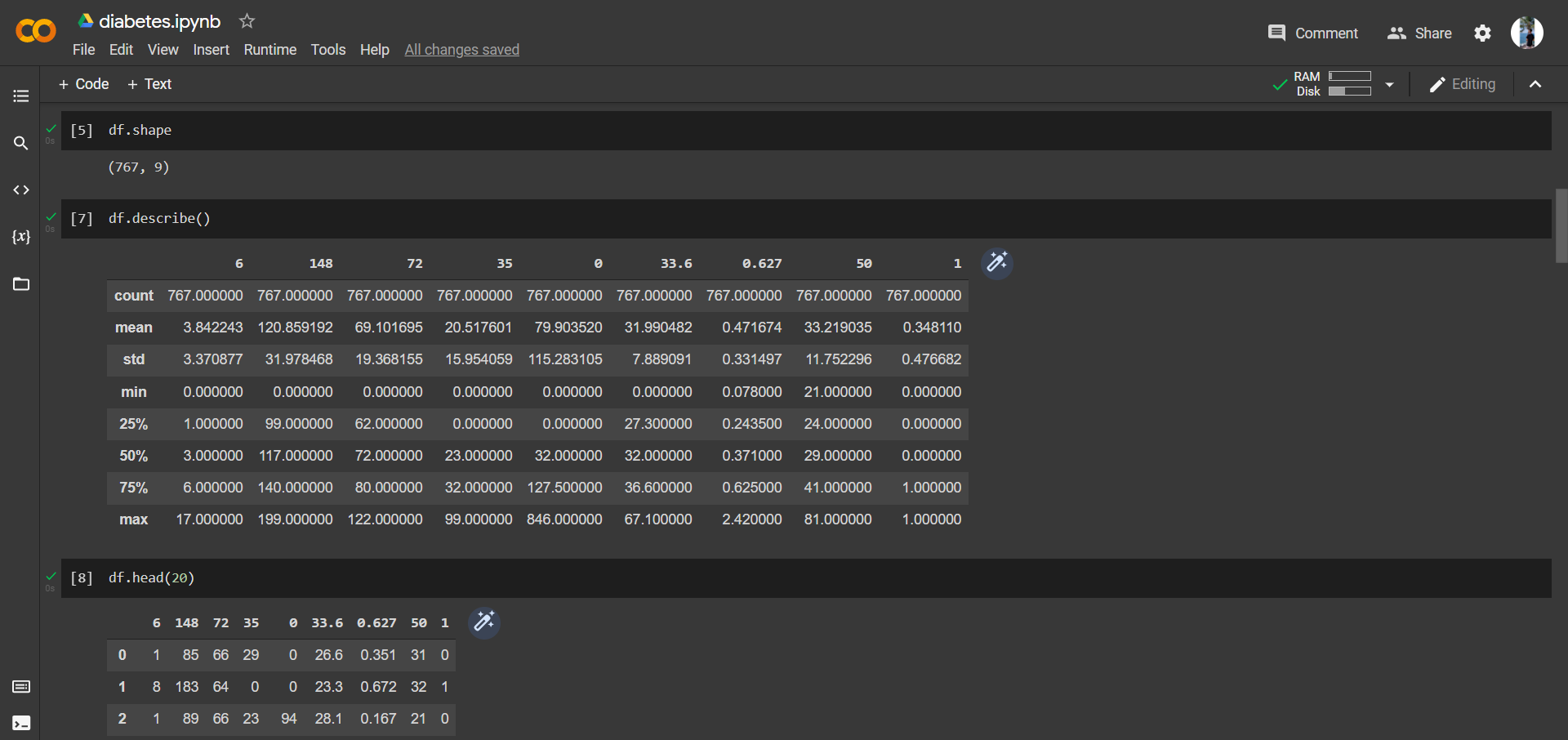
● fillna()

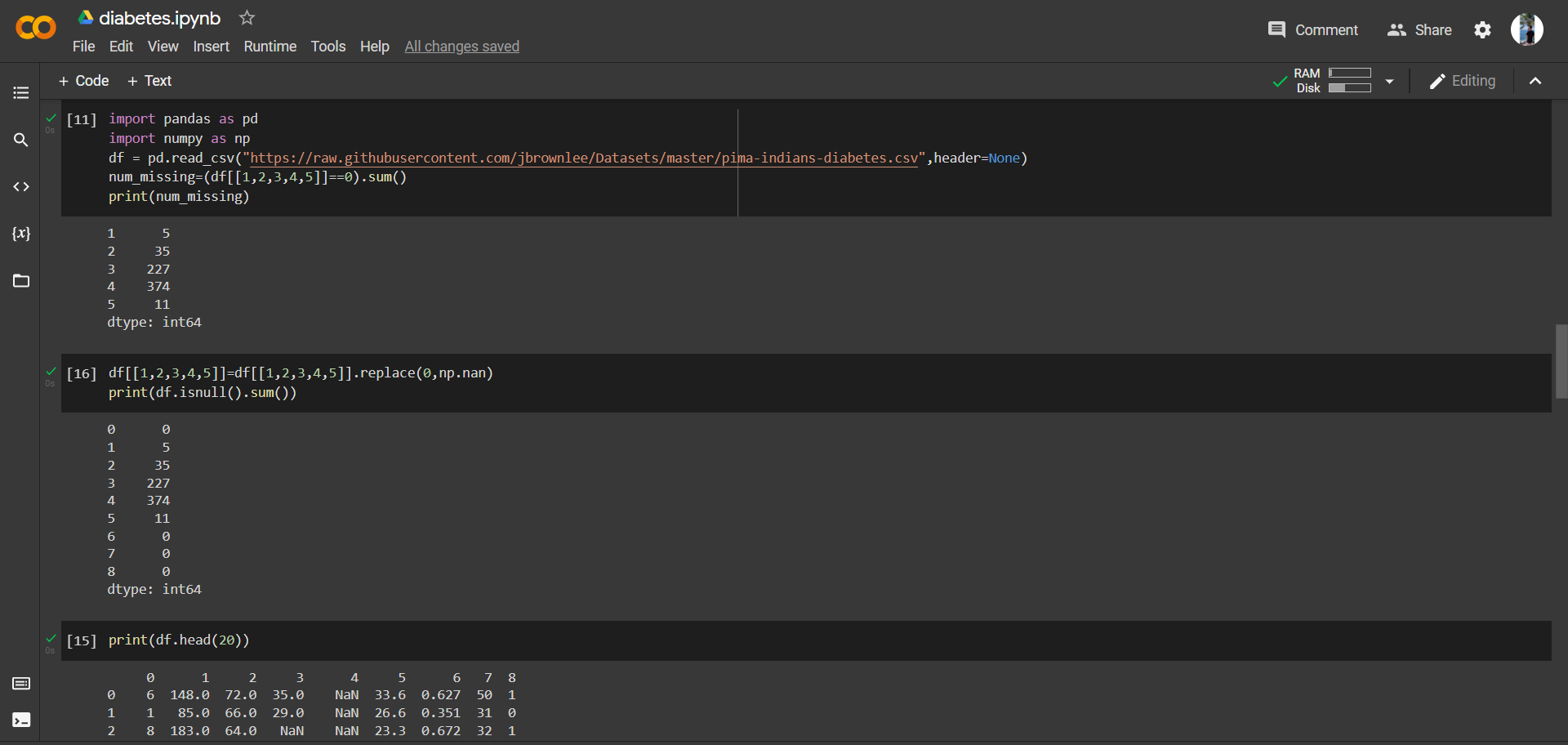
● replace

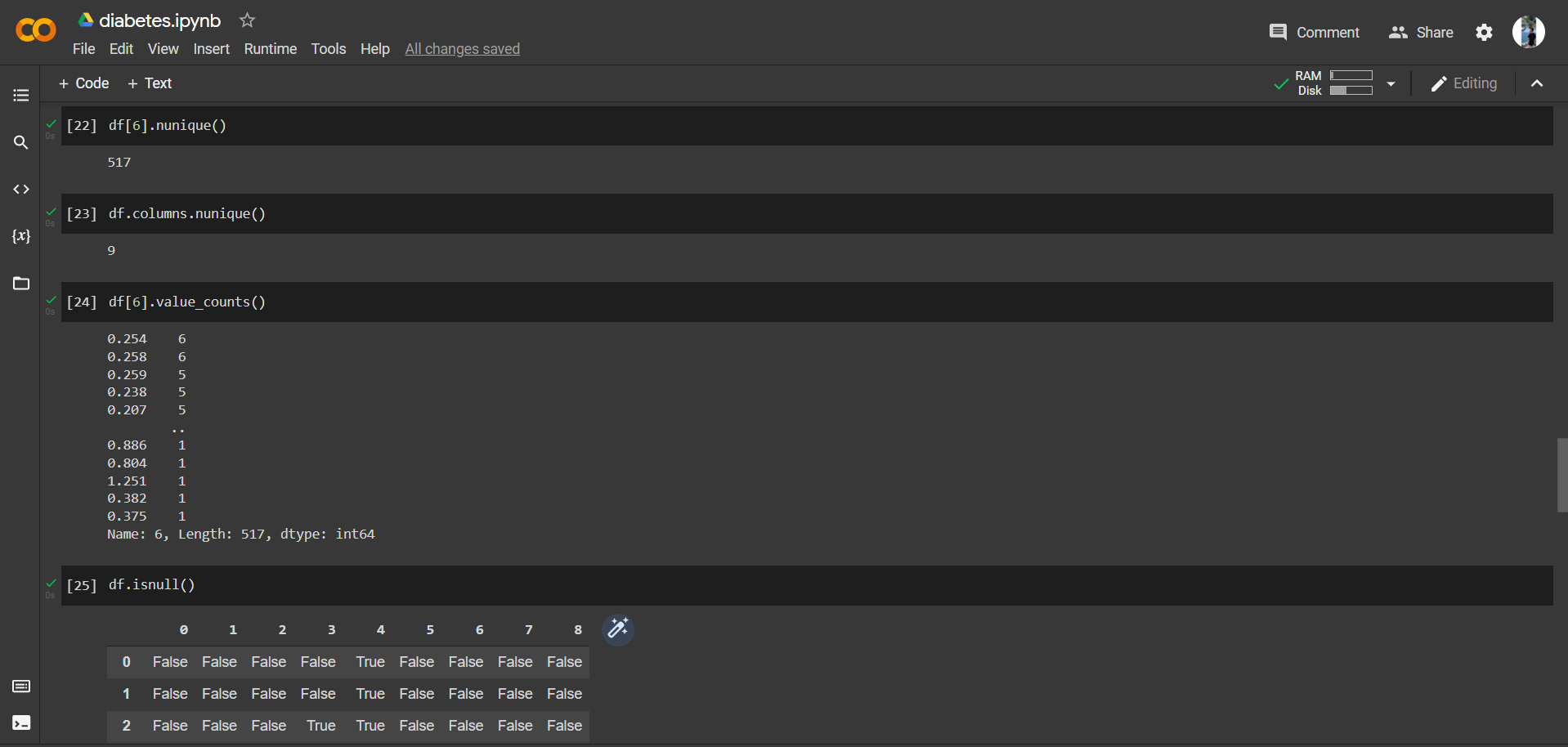
● interpolate()

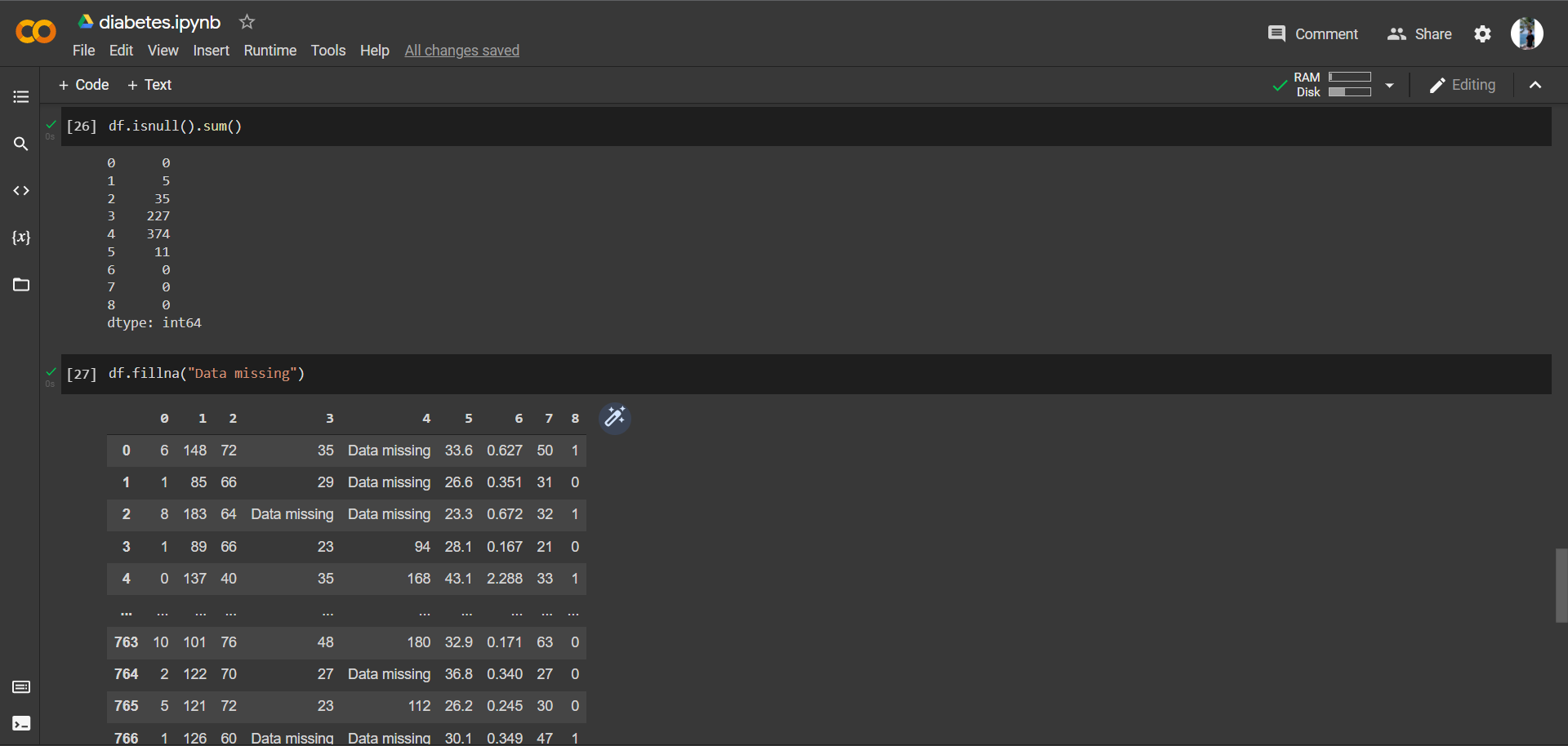
Diabetes dataset eg

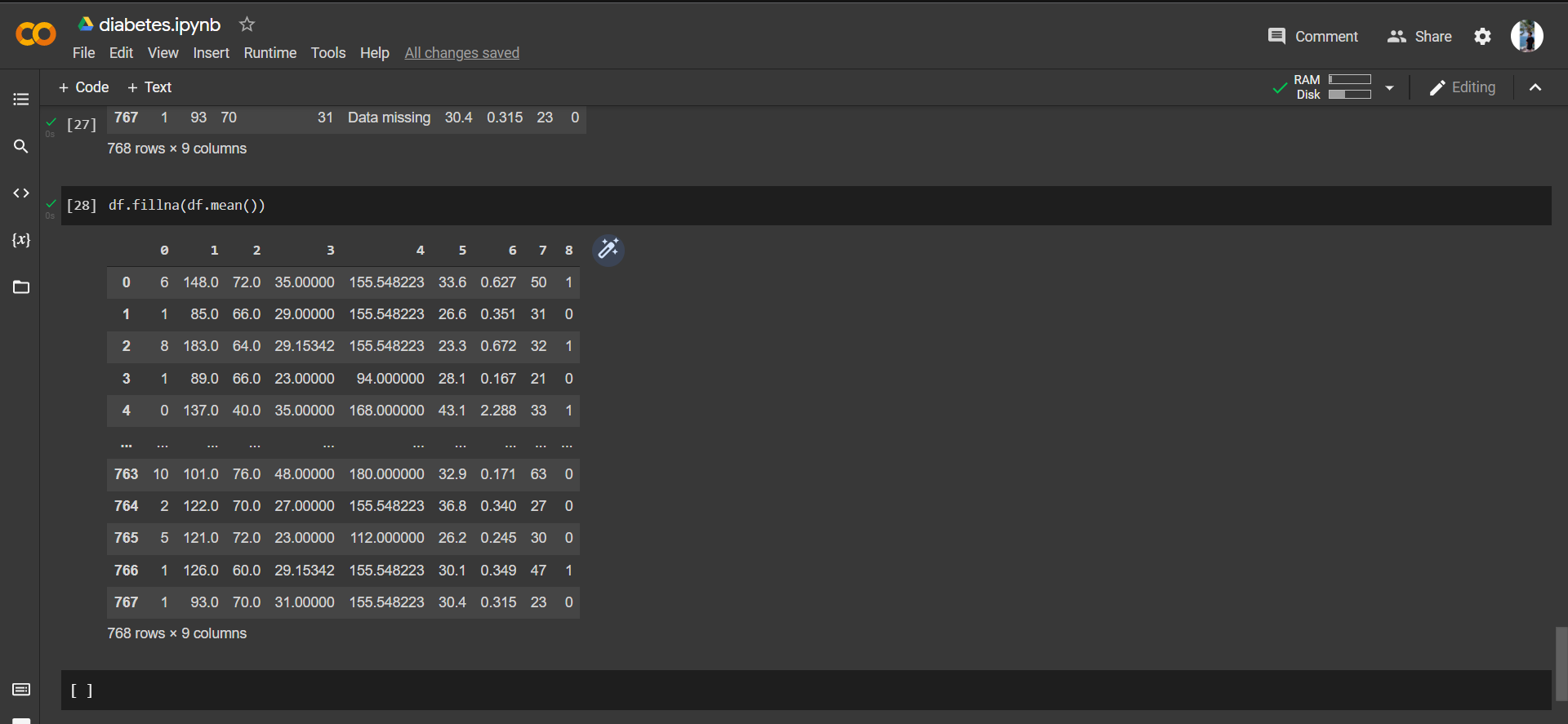








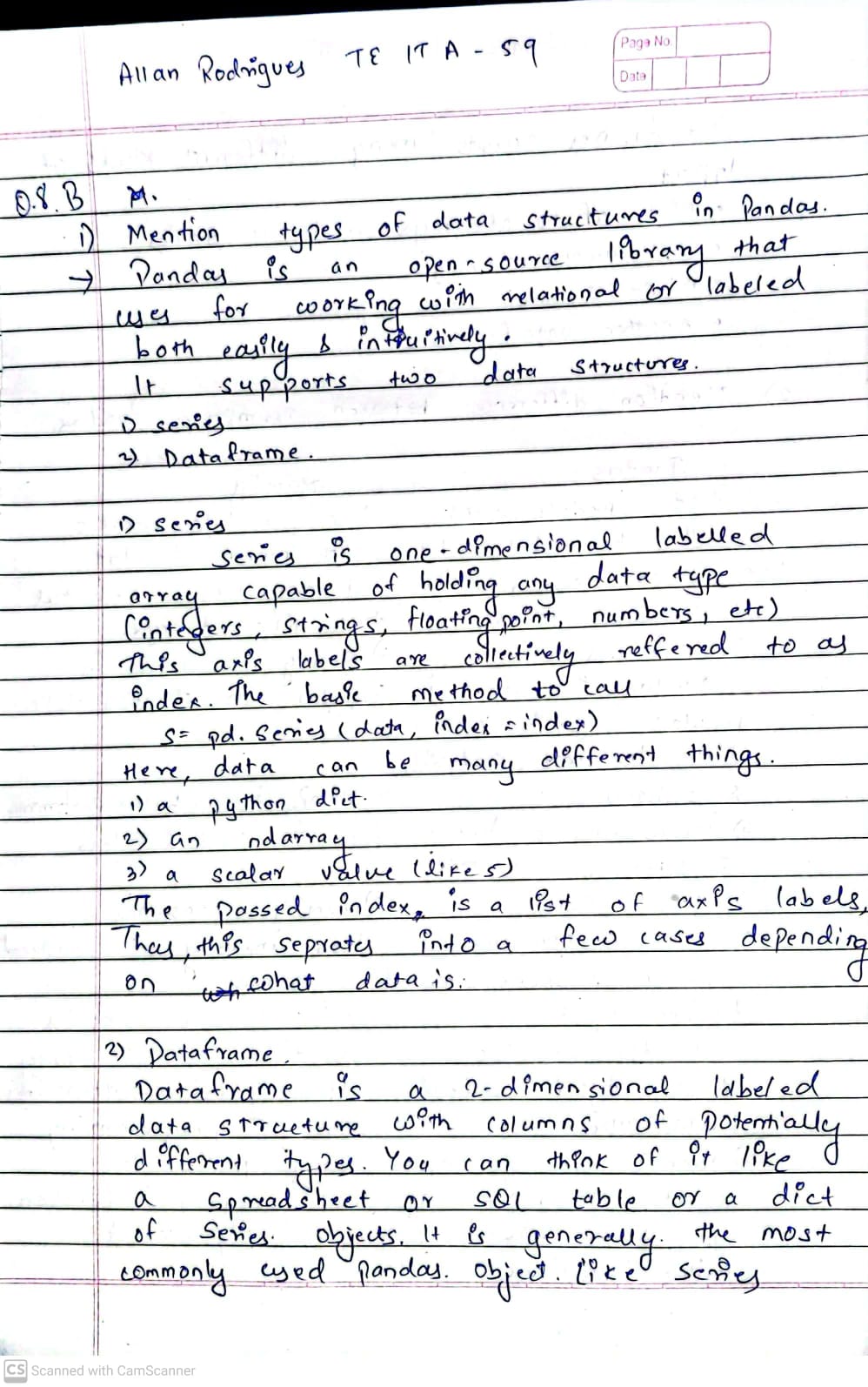


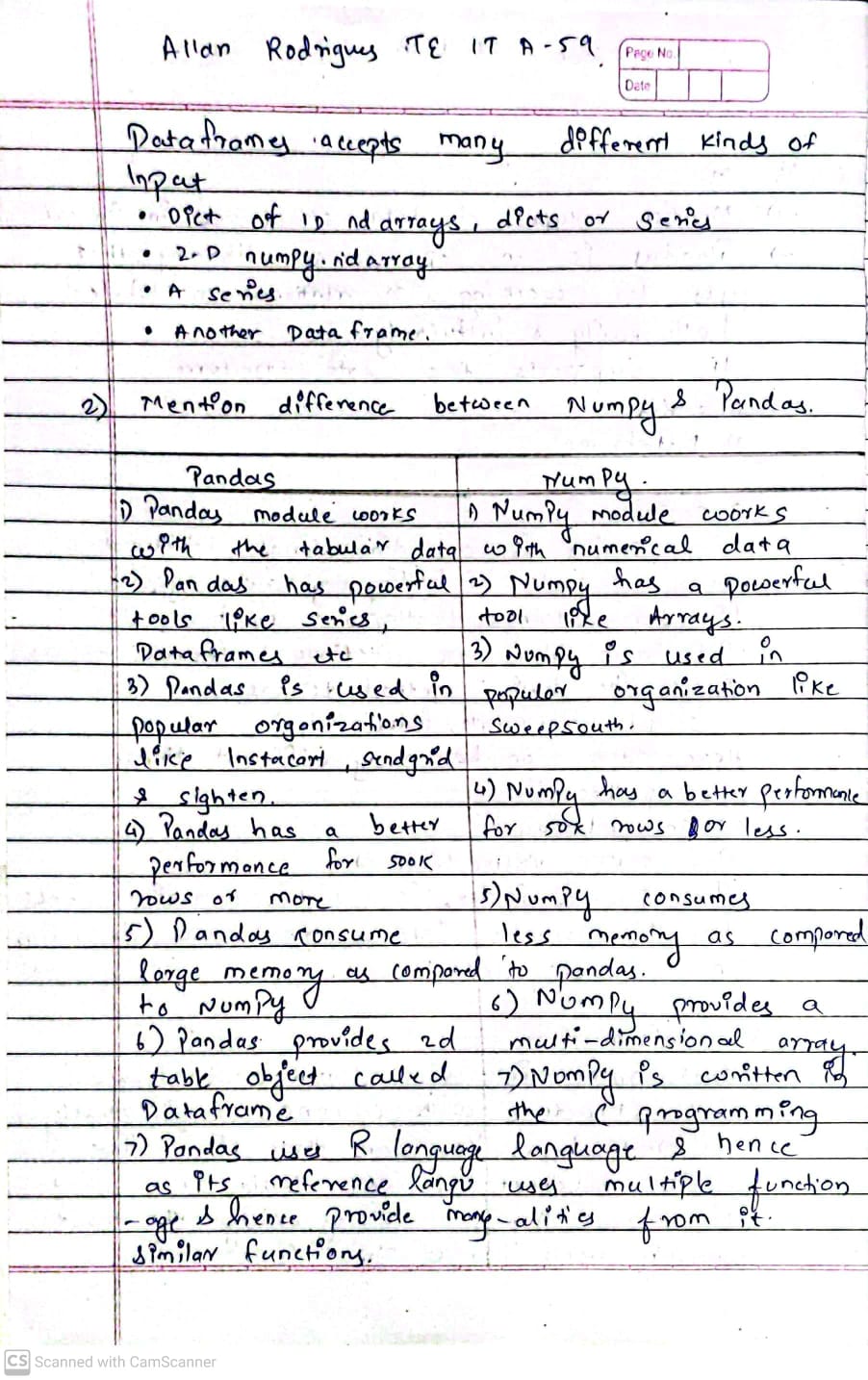


**B. Questions:**

Mention types of data structures in Pandas.

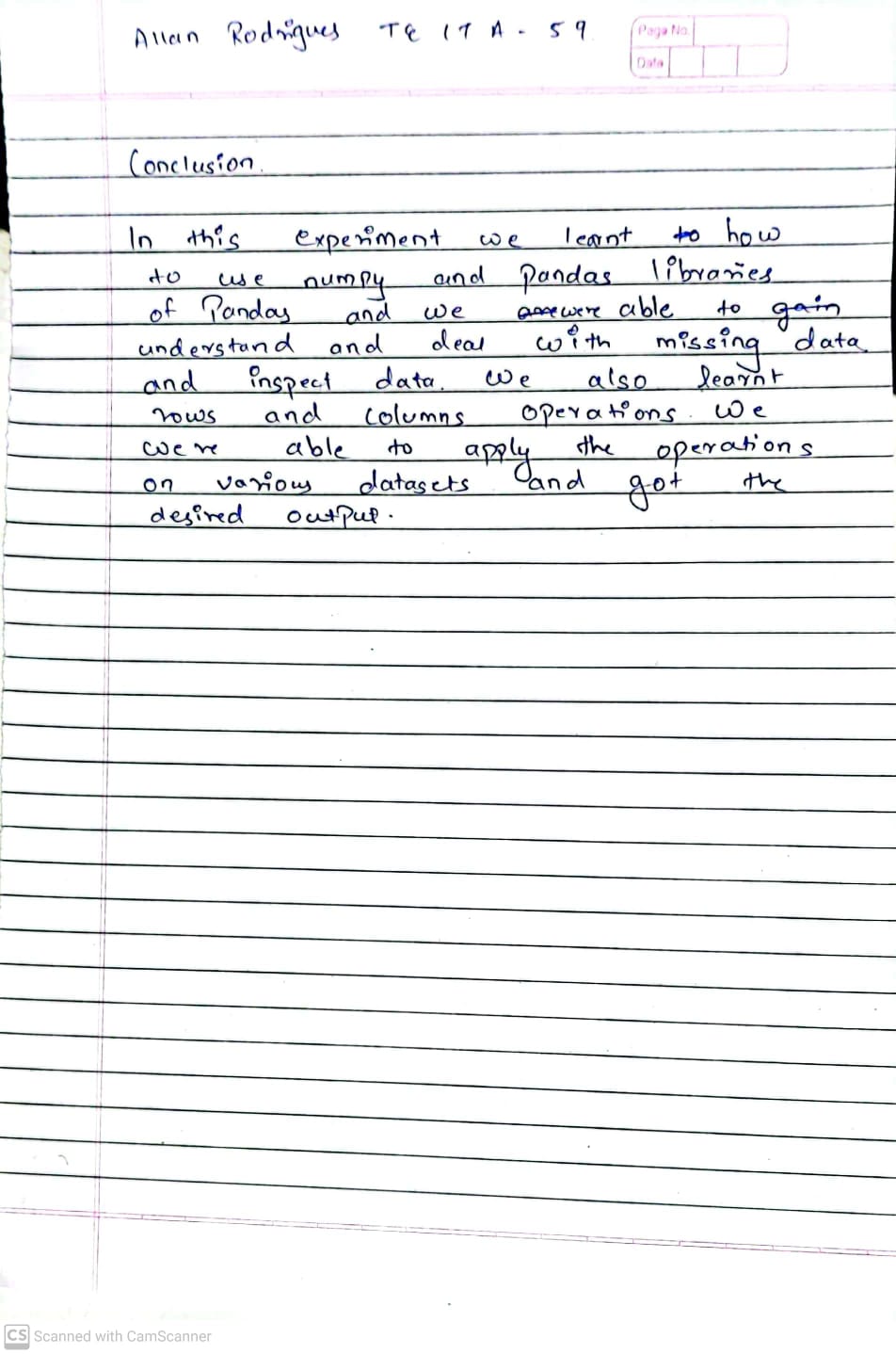
Mention difference between Numpy and Pandas.





**C. Conclusion:**

Write the significance of the topic studied in the experiment.



**D. References:**

1. [How to Handle Missing Data with Python (machinelearningmastery.com)](https://machinelearningmastery.com/handle-missing-data-python/)
2. <https://www.w3schools.com/python/pandas>
3. <https://www.geeksforgeeks.org/difference-between-pandas-vs-numpy/>

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