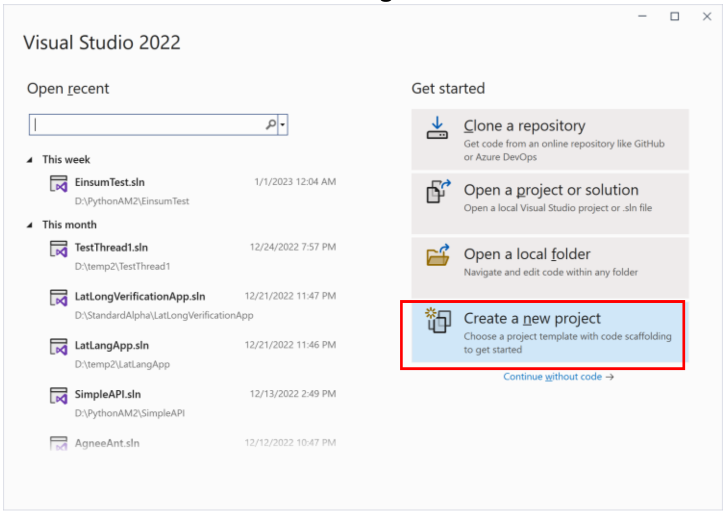
Assignment #1 – CPSC 552
DataFrames in Python

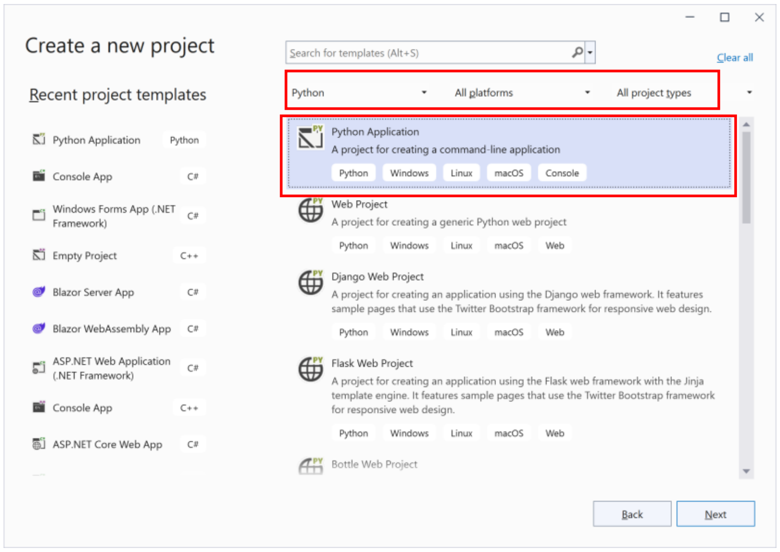
Pandas library in Python provides a class to store and process two dimensional data (like an Excel sheet) called "DataFrame". We can read an Excel file into a dataframe, or lists, dictionaries, and other dataframes can be put in a dataframe.
The general form of the dataframe creation is:

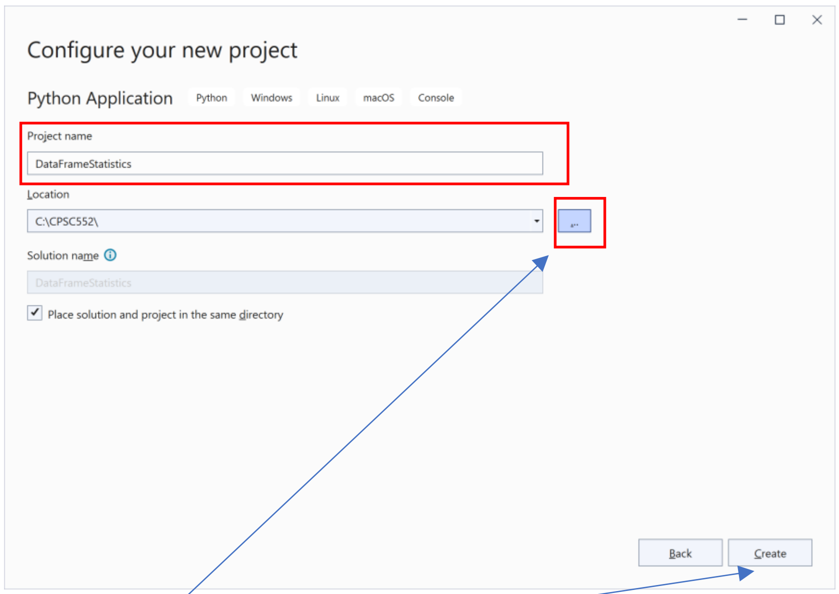
pandas.DataFrame(data, index, columns, dtype, copy)

data – it can be a numpy ndarray, list, dictionary or another dataframe.
columns - default sis - np.arange(n) i.e., column 0, column 1,..,column n-1. Named columns can be passed as well.
dtype - data type of each column.
copy - this is used for copying of data, the default is false.

<u>**Example**</u>: Launch Visual Studio and create a Python application called **DataFrameStatistics** as shown below. Selections to be made are shown in red rectangles.


Visual Studio 2022
Open recent
This week
EinsumTest.sln
This month
TestThread1.sln
LatLongVerificationApp.sln
LatLangApp.sln
SimpleAPI.sln
AgneeAnt.sln
Get started
Clone a repository
Get code from an online repository like GitHub or Azure DevOps
Open a project or solution
Open a local Visual Studio project or .sln file
Open a local folder
Navigate and edit code within any folder
Create a new project
Choose a project template with code scaffolding to get started
Continue without code →


Create a new project
Search for templates (Alt+S)
Clear all
Recent project templates
Python Application Python
Console App C#
Windows Forms App (.NET Framework) C#
Empty Project C++
Blazor Server App C#
Blazor WebAssembly App C#
ASP.NET Web Application (.NET Framework) C#
Console App C++
ASP.NET Core Web App C#
Python
All platforms
All project types
Python Application
A project for creating a command-line application
Python Windows Linux macOS Console
Web Project
A project for creating a generic Python web project
Python Windows Linux macOS Web
Django Web Project
A project for creating an application using the Django web framework. It features sample pages that use the Twitter Bootstrap framework for responsive web design.
Python Windows Linux macOS Web
Flask Web Project
A project for creating an application using the Flask web framework with the Jinja template engine. It features sample pages that use the Twitter Bootstrap framework for responsive web design.
Python Windows Linux macOS Web
Bottle Web Project
Back
Next



Configure your new project

Python Application — Python, Windows, Linux, macOS, Console

Project name: DataFrameStatistics

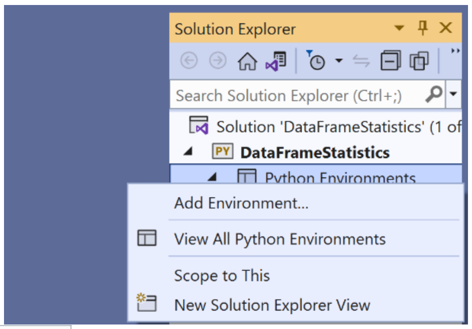
Location: C:\CPSC552\

Solution name: DataFrameStatistics

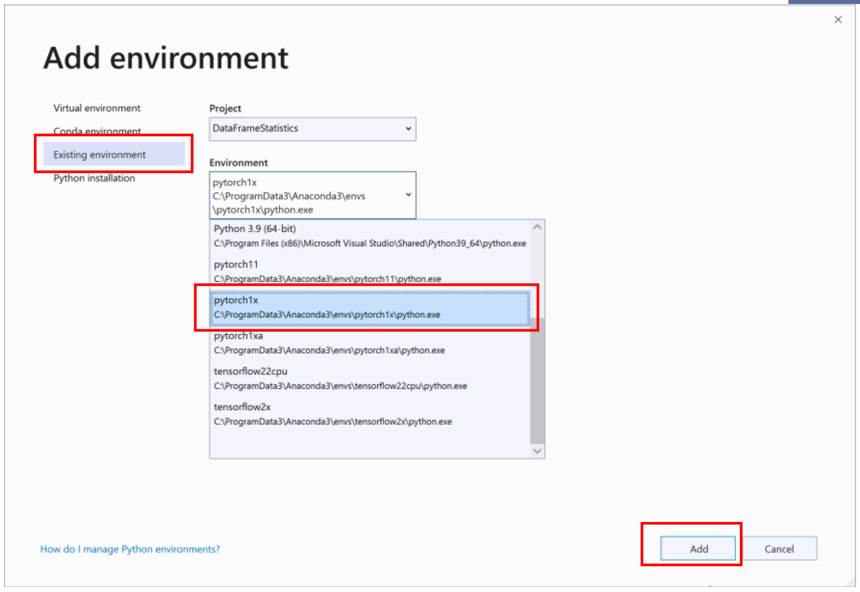
☑ Place solution and project in the same directory

Click on the three dotted button to create a folder called CPSC552 to store the project on the C drive. Once you click the Create button, the project will be created for you.

Next, you need to set the proper Python environment e.g., **Pytorch1x** in Visual Studio. Right click on Python Environments (under Solution explorer) and choose Add Environment.



Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'DataFrameStatistics' (1 of
DataFrameStatistics
Python Environments
Add Environment...
View All Python Environments
Scope to This
New Solution Explorer View



Add environment

Project: DataFrameStatistics

Environment: pytorch1x C:\ProgramData3\Anaconda3\envs\pytorch1x\python.exe

Python 3.9 (64-bit) C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe

pytorch11 C:\ProgramData3\Anaconda3\envs\pytorch11\python.exe

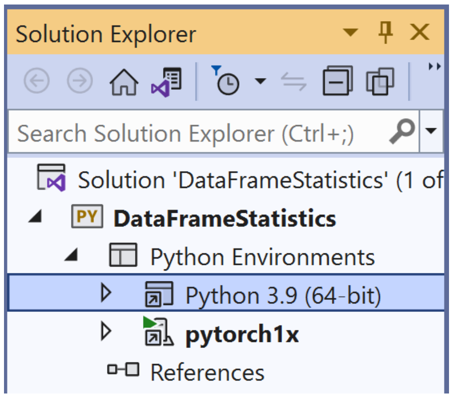
pytorch1x C:\ProgramData3\Anaconda3\envs\pytorch1x\python.exe

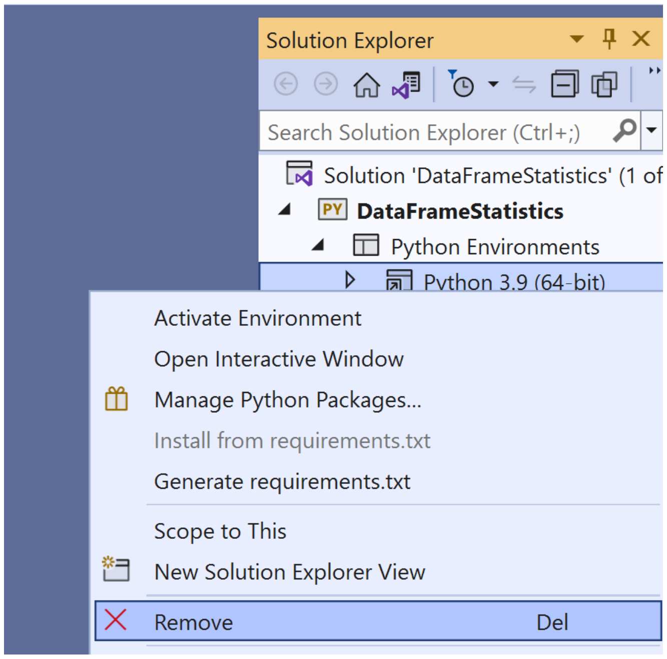
pytorch1xa C:\ProgramData3\Anaconda3\envs\pytorch1xa\python.exe

tensorflow22cpu C:\ProgramData3\Anaconda3\envs\tensorflow22cpu\python.exe

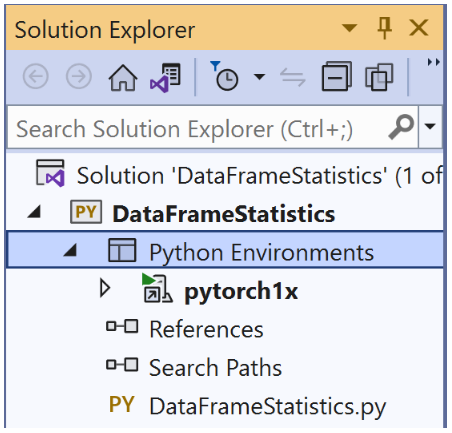
tensorflow2x C:\ProgramData3\Anaconda3\envs\tensorflow2x\python.exe

If multiple environments show up under the environments in solution explorer, remove the environment other than “pytorch1x” by right clicking on it and choosing “remove environment”.



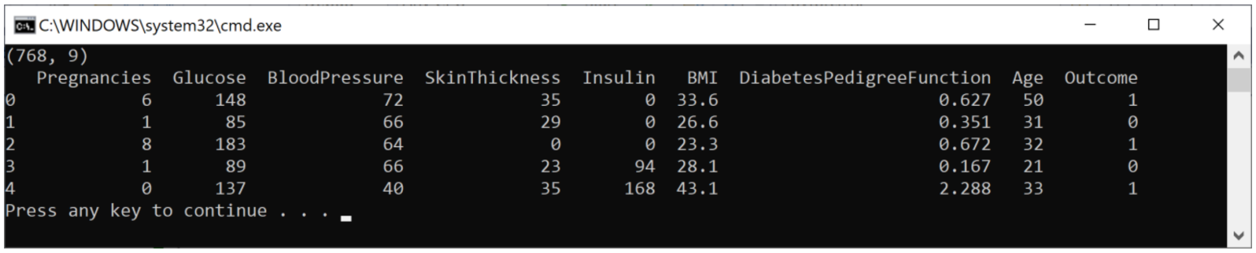

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'DataFrameStatistics' (1 of
DataFrameStatistics
Python Environments
Python 3.9 (64-bit)
Activate Environment
Open Interactive Window
Manage Python Packages...
Install from requirements.txt
Generate requirements.txt
Scope to This
New Solution Explorer View
Remove
Del

Removing a Python environment only removes it from the project and not your computer so it is safe to remove an environment. Make sure that only one environment e.g., “pytorch1x” is shown in bold under environments in Visual Studio (see below).



We will analyze the statistics in the PimaIndians diabetes dataset. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. All patients here are females at least 21 years old of Pima Indian heritage.

There are 768 rows and 9 columns. The last column called outcome indicates if the person has diabetes or not.

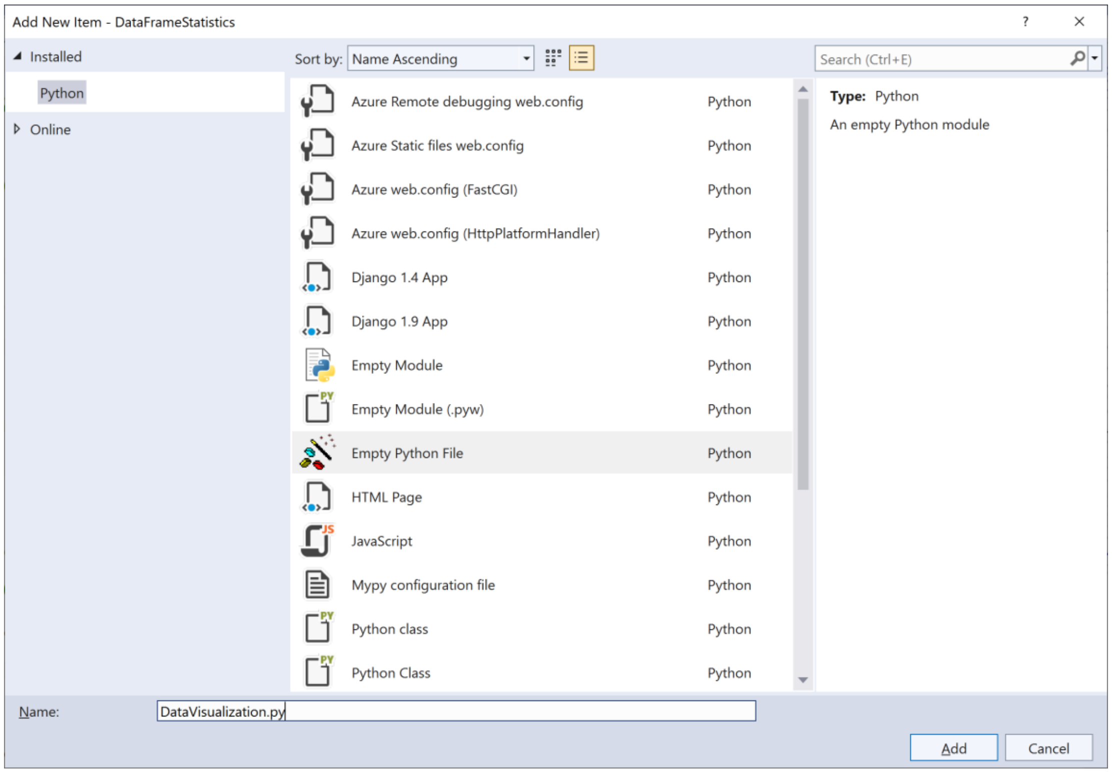


```
(768, 9)
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  DiabetesPedigreeFunction  Age  Outcome
0            6      148             72             35        0  33.6                     0.627   50        1
1            1       85             66             29        0  26.6                     0.351   31        0
2            8      183             64              0        0  23.3                     0.672   32        1
3            1       89             66             23       94  28.1                     0.167   21        0
4            0      137             40             35      168  43.1                     2.288   33        1
Press any key to continue . . .
```

Type the following code in DataFrameStatistics.py file. Type up to each print statement and then run the program to examine and understand the output it generates. You will have to change the location of the csv file to the folder where you have downloaded the file from the kiwi web site (unzip the dataset zip file after downloading from the kiwi web site).

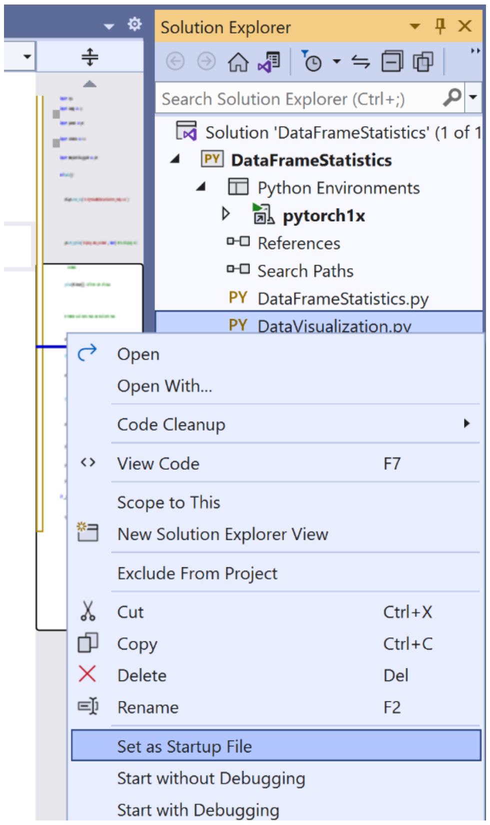
```python
import sys
import numpy as np
import pandas as pd


def main():
    df=pd.read_csv('D:/PythonAM/Data/diabetes_temp.csv')
    pd.set_option('display.max_columns', None) # to display all columns
    print(df.shape)

    print(df.head())  # first set of rows
    print(df.tail())  # last set of rows

    print(df.info())  # null, not null and data type of each column

    print(df.dtypes)  # data types of each column

    print("Null values\n",df.isna().sum())
    print("Duplicate values", df.duplicated().sum())  # dulicate values

    # remove null data rows and dulicate rows
    dfn = df.dropna()
    print(dfn.shape)
    df_clean = dfn.drop_duplicates()
    print(df_clean.shape)

    column_headers = list(df.columns.values)
    print("Column Headings :", column_headers)

    # last column is outcome, lets remove it from the columns list
    # so that we can compute correlation between the data columns
    data_columns = column_headers[0:-1]
    print(data_columns)

    data_corr = df_clean[data_columns].corr() # correlation between columns
    print(data_corr)

    # determine the highest correlations for a feature
    corr_data = df_clean.corr().abs()
    cbmi = corr_data["BMI"]
    corrbmi = cbmi.sort_values(ascending = False)
    print("sorted correlations--------")
    print(corrbmi)

    print(df_clean.describe())

    print(df_clean.nunique())

    # print unique values of Pregnancies
    print(df_clean['Pregnancies'].unique())

    # combining columns from one dataframe to another new dataframe
    seriesBMI = df["BMI"]
    seriesGlucose = df["Glucose"]
    seriesOutcome = df["Outcome"]
    dfnew = pd.concat([seriesBMI,seriesGlucose, seriesOutcome], axis=1)
    print(dfnew.head)

    #---filtering or removing rows from a dataframe based on criteria
    dffiltered = dfnew[dfnew['Glucose']<= 180]
    print(len(dffiltered))
    print(dffiltered.head())

    # change Outcome colum 1 and 0 classes to 1 and 2
    dfnew['Outcome'] = dfnew['Outcome'].map(lambda x: 1 if x ==1 else 2)
    print(dfnew.head)

    # access a particular row, column data in a dataframe
    d1 = dfnew.iloc[25,1]  # glucose value in row 25
    print(d1)

    # select 10-13 rows into a new dataframe
    d2 = dfnew.iloc[10:14,:]
    print(d2.head())

    # select 10-13 rows into a new dataframe and reset index
    d3 = dfnew.iloc[10:14,:].reset_index()
    print(d3.head())

    # select 10-13 rows into a new dataframe and reset index
    # drop=True to avoid old index being kept
    d3 = dfnew.iloc[10:14,:].reset_index(drop=True)
    print(d3.head())

if __name__ == "__main__":
    sys.exit(int(main() or 0))
```

The above code shows how data can be read into a dataframe and do cleanup and determine basic statistics on the data. We also try to visualize the data in order to better understand the data distributions. Add another file to the project called DataVisualization.py by right clicking on the project name and choosing add new item as shown below.



Name: DataVisualization.py

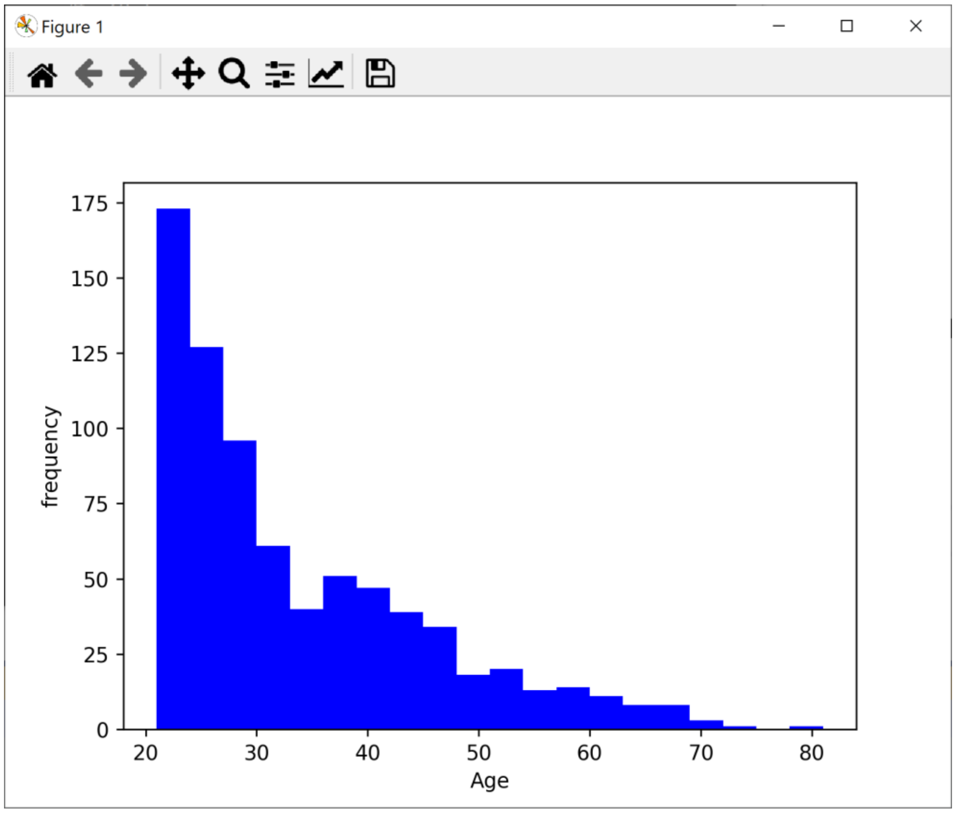
Set the DataVisulaization.py as the startup file by right clicking on it in the solution explorer and choosing “set as startup file” as shown below.



Type the following code in DataVisualization.py. Type up to each plt.show() statement and then run the program to see the visualization it produces.

```
import sys
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# pip install seaborn

def main():
    df=pd.read_csv('D:/PythonAM/Data/diabetes_temp.csv')
    pd.set_option('display.max_columns', None) # to display all columns
    print(df.head())  # first set of rows
    # remove null data rows and dulicate rows
    dfn = df.dropna()
    print(dfn.shape)
    dfc = dfn.drop_duplicates()
    print(dfc.shape)

    dfc.Age.plot(color="blue",kind="hist", bins=20)
    plt.xlabel("Age")
    plt.ylabel("frequency")
    plt.show()

    dfc.hist(column='Age', by='Outcome') # histogram by group
    plt.show()

    df2= dfc[["Age","Glucose"]]
    df2.plot(kind='hist',
        alpha=0.7,
        bins=30,
        title='Histogram Of Age,Glucose',
        rot=45,
        grid=True,
        figsize=(12,8),
        fontsize=15,
        color=['#ff0000', '#00ff00'])
    plt.xlabel('Age/Glucose')
    plt.ylabel("Count");
    plt.show()

    sns.countplot(x="Outcome", data=dfc)
    plt.show()
    print(dfc.Outcome.value_counts())  # outcome counts

    sizes=dfc.Pregnancies.value_counts().values[0:9]
    labels=dfc.Pregnancies.value_counts().index[0:9]
    colors=["green","pink","yellow","purple","grey","blue","plum","orange",
"red"]
    plt.pie(sizes,data=dfc,labels=labels,colors=colors,radius=1.5)
    plt.show()

    df2=(dfc.columns[0:-1])  # remove outcomes column
    dfc.hist(df2,bins=50, figsize=(20,15))
    plt.show()

    sns.heatmap(dfc[df2].corr(),annot=True)
    plt.show()

if __name__ == "__main__":
    sys.exit(int(main() or 0))
```

The different Visualizations that appear are shown below (you will need to x out each plot in order to see the next plot).



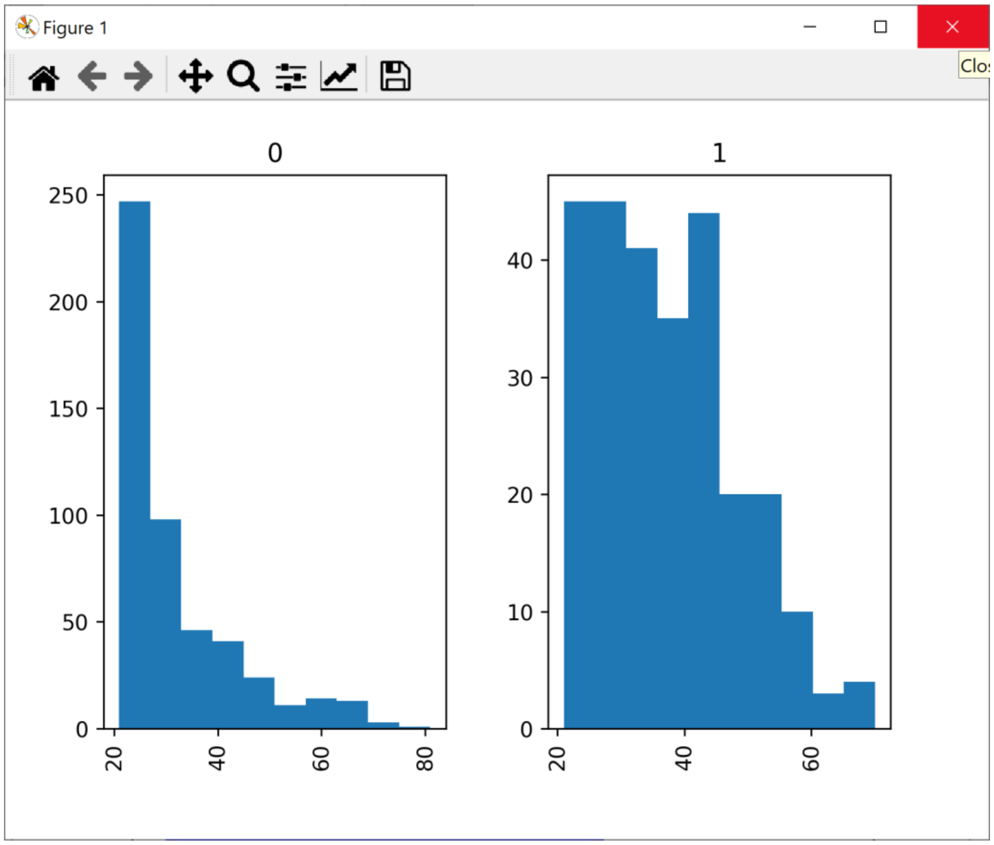


Figure 1
Clos
0
1
250
200
150
100
50
0
20
40
60
80
40
30
20
10
0
20
40
60

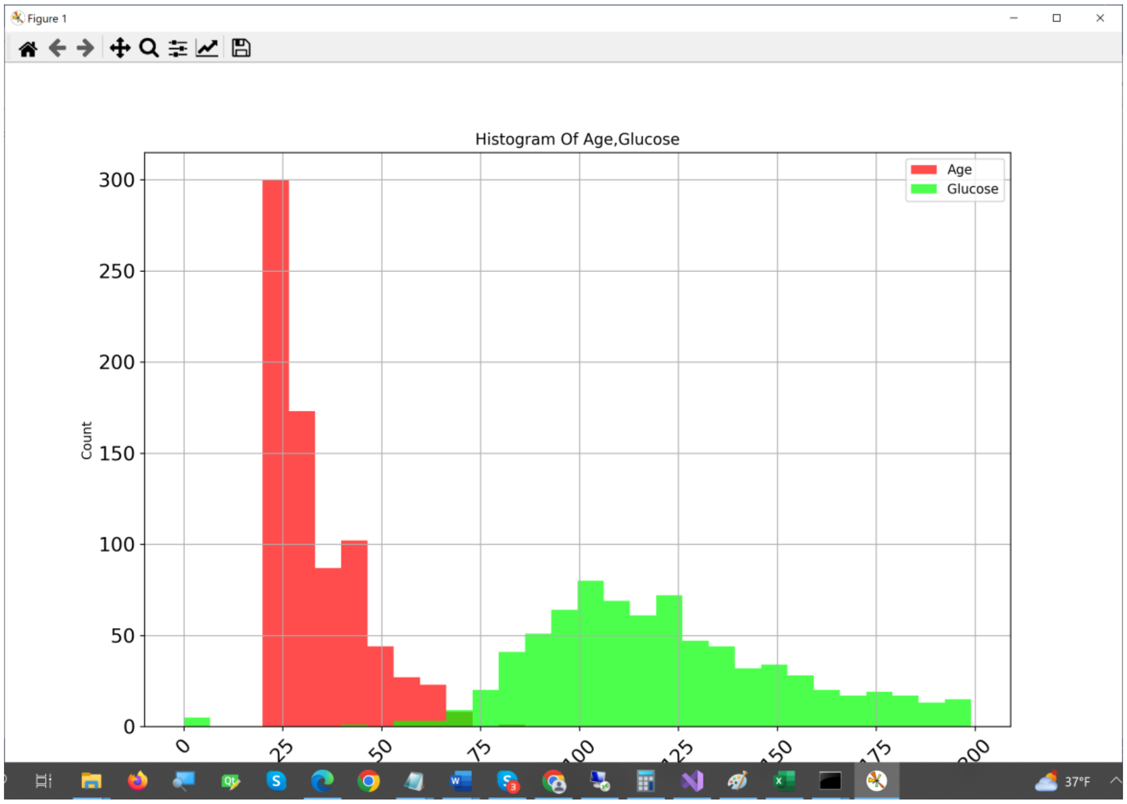


Figure 1
Histogram Of Age,Glucose
Age
Glucose
Count
300
250
200
150
100
50
0
0
25
50
75
100
125
150
175
200
37°F

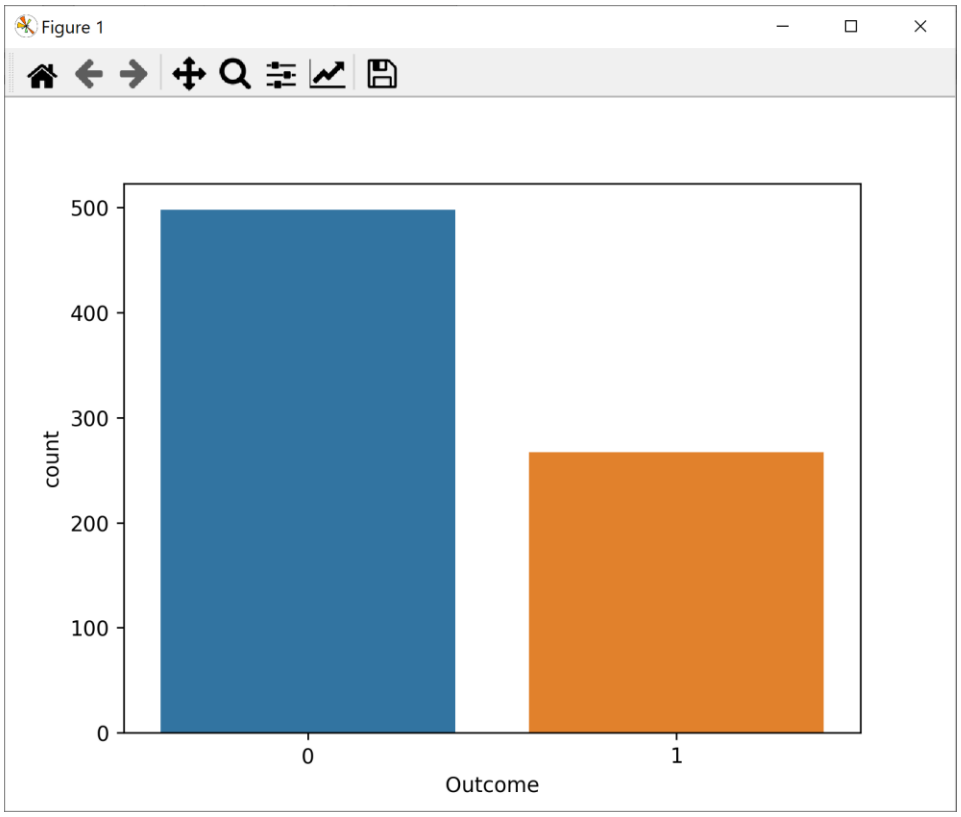


Figure 1
500
400
300
200
100
0
count
0
1
Outcome

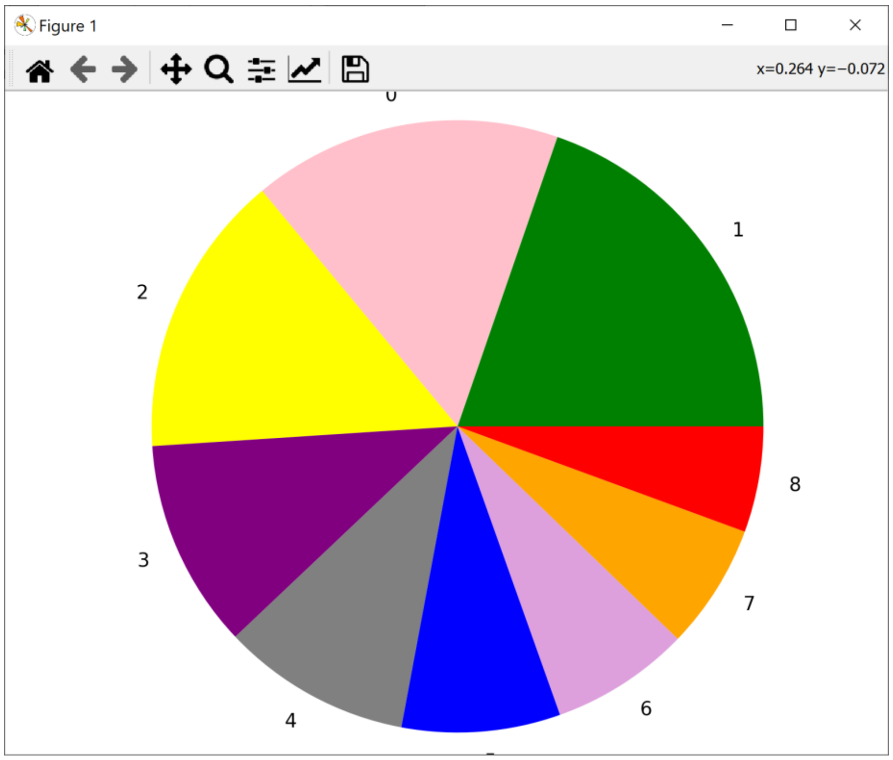
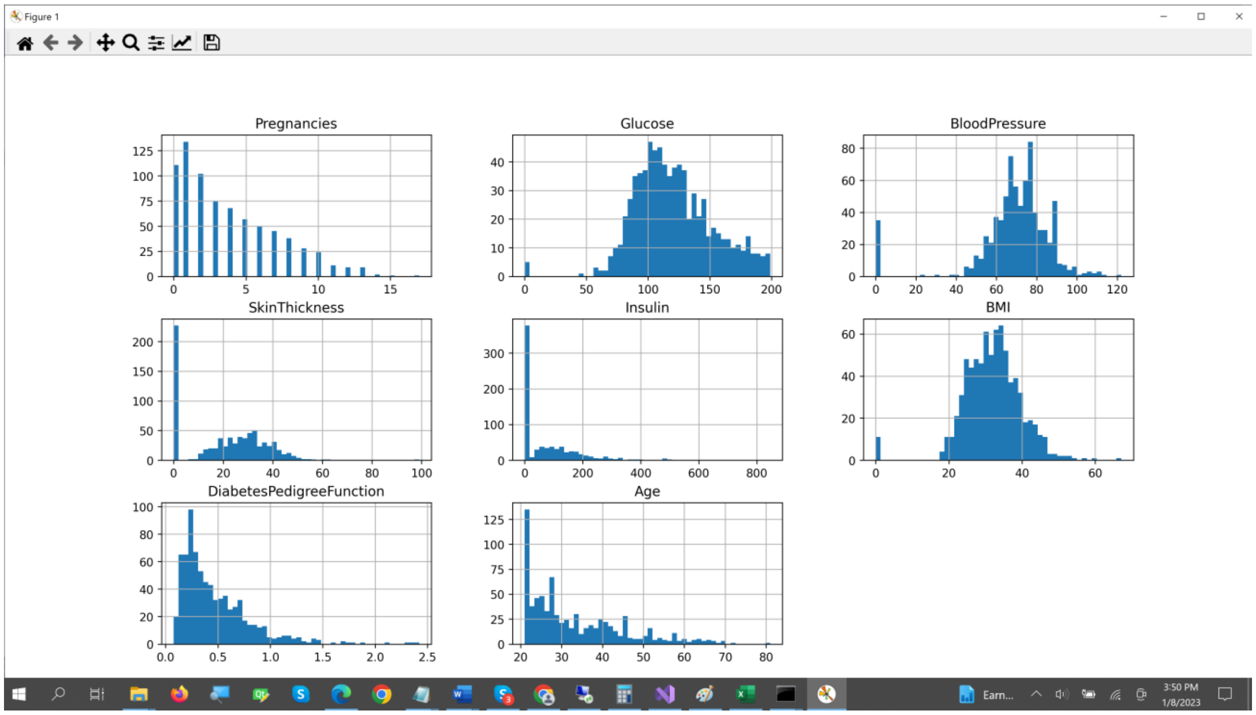
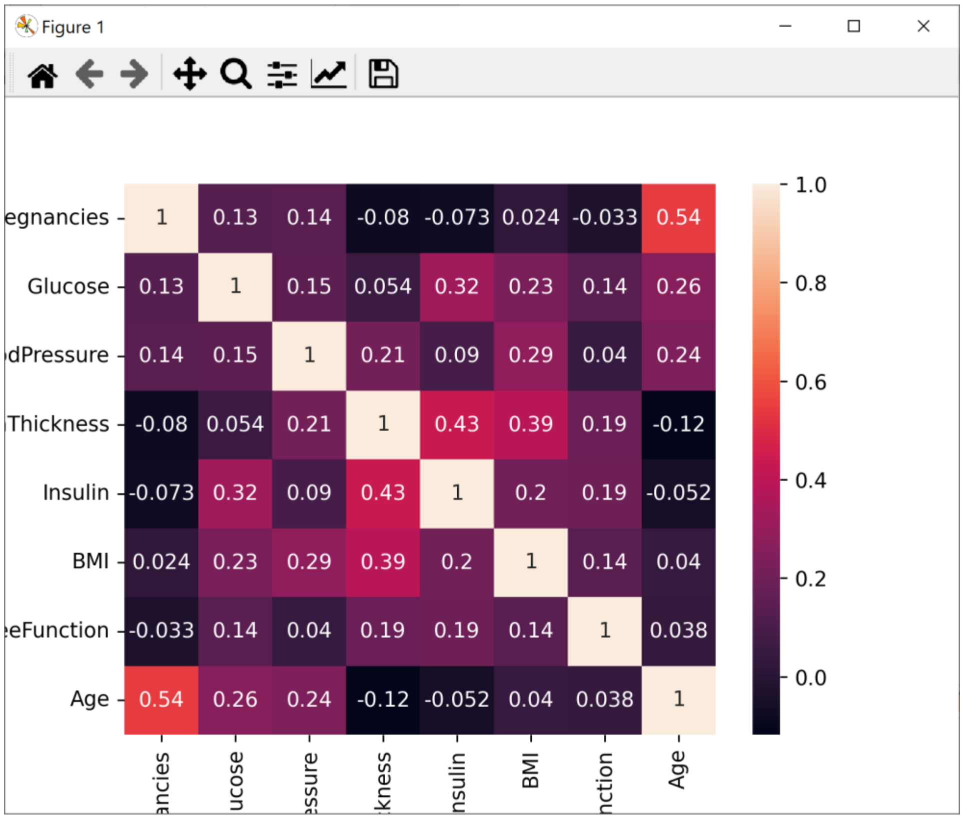


Figure 1
x=0.264 y=−0.072
1
2
3
4
6
7
8





	ancies	ucose	essure	kness	nsulin	BMI	nction	Age
egnancies	1	0.13	0.14	-0.08	-0.073	0.024	-0.033	0.54
Glucose	0.13	1	0.15	0.054	0.32	0.23	0.14	0.26
dPressure	0.14	0.15	1	0.21	0.09	0.29	0.04	0.24
Thickness	-0.08	0.054	0.21	1	0.43	0.39	0.19	-0.12
Insulin	-0.073	0.32	0.09	0.43	1	0.2	0.19	-0.052
BMI	0.024	0.23	0.29	0.39	0.2	1	0.14	0.04
eFunction	-0.033	0.14	0.04	0.19	0.19	0.14	1	0.038
Age	0.54	0.26	0.24	-0.12	-0.052	0.04	0.038	1