from PyQt5 import QtCore, QtGui, QtWidgets

from subprocess import call

import webbrowser

url\_try = "https://colab.research.google.com/"

url\_plot = url = "http://localhost:8888/notebooks/Plot.ipynb"

url\_quiz = "http://localhost:8888/notebooks/Quiz.ipynb"

chrome\_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(1000, 750)

palette = QtGui.QPalette()

brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)

brush = QtGui.QBrush(QtGui.QColor(109, 109, 109))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)

brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.WindowText, brush)

brush = QtGui.QBrush(QtGui.QColor(109, 109, 109))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)

brush = QtGui.QBrush(QtGui.QColor(120, 120, 120))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.WindowText, brush)

brush = QtGui.QBrush(QtGui.QColor(240, 240, 240))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)

MainWindow.setPalette(palette)

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setBold(True)

font.setWeight(75)

MainWindow.setFont(font)

icon = QtGui.QIcon()

icon.addPixmap(QtGui.QPixmap("res/ico/MyPY.ico"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

MainWindow.setWindowIcon(icon)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setStyleSheet("background-color: rgb(35, 35, 35);")

self.centralwidget.setObjectName("centralwidget")

self.label\_title = QtWidgets.QLabel(self.centralwidget)

self.label\_title.setGeometry(QtCore.QRect(340, 10, 301, 81))

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setPointSize(36)

font.setBold(True)

font.setWeight(75)

self.label\_title.setFont(font)

self.label\_title.setStyleSheet("color: rgb(255, 255, 255);")

self.label\_title.setObjectName("label\_title")

self.label\_tagline = QtWidgets.QLabel(self.centralwidget)

self.label\_tagline.setGeometry(QtCore.QRect(338, 100, 401, 31))

font = QtGui.QFont()

font.setFamily("Lucida Handwriting")

font.setPointSize(14)

self.label\_tagline.setFont(font)

self.label\_tagline.setStyleSheet("color: rgb(255, 255, 255);")

self.label\_tagline.setObjectName("label\_tagline")

self.line = QtWidgets.QFrame(self.centralwidget)

self.line.setGeometry(QtCore.QRect(7, 140, 991, 20))

self.line.setStyleSheet("color : rgb(135, 135, 135)")

self.line.setFrameShape(QtWidgets.QFrame.HLine)

self.line.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line.setObjectName("line")

self.label\_hello = QtWidgets.QLabel(self.centralwidget)

self.label\_hello.setGeometry(QtCore.QRect(10, 160, 251, 31))

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.label\_hello.setFont(font)

self.label\_hello.setStyleSheet("color: rgb(255, 255, 255);")

self.label\_hello.setObjectName("label\_hello")

self.label\_learn = QtWidgets.QLabel(self.centralwidget)

self.label\_learn.setGeometry(QtCore.QRect(10, 200, 291, 21))

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setPointSize(10)

font.setBold(True)

font.setWeight(75)

self.label\_learn.setFont(font)

self.label\_learn.setStyleSheet("color: rgb(255, 255, 255);")

self.label\_learn.setObjectName("label\_learn")

self.pushButton\_bar = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_bar.setGeometry(QtCore.QRect(10, 240, 93, 28))

self.pushButton\_bar.setStyleSheet("background-color: rgb(109, 109, 109);\n"

"")

self.pushButton\_bar.setObjectName("pushButton\_bar")

self.pushButton\_box = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_box.setGeometry(QtCore.QRect(140, 240, 93, 28))

self.pushButton\_box.setStyleSheet("background-color: rgb(109, 109, 109);")

self.pushButton\_box.setObjectName("pushButton\_box")

self.pushButton\_freq = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_freq.setGeometry(QtCore.QRect(270, 240, 141, 28))

self.pushButton\_freq.setStyleSheet("background-color: rgb(109, 109, 109);")

self.pushButton\_freq.setObjectName("pushButton\_freq")

self.pushButton\_hist = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_hist.setGeometry(QtCore.QRect(450, 240, 93, 28))

self.pushButton\_hist.setStyleSheet("background-color: rgb(109, 109, 109);")

self.pushButton\_hist.setObjectName("pushButton\_hist")

self.pushButton\_line = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_line.setGeometry(QtCore.QRect(580, 240, 93, 28))

self.pushButton\_line.setStyleSheet("background-color: rgb(109, 109, 109);")

self.pushButton\_line.setObjectName("pushButton\_line")

self.pushButton\_pie = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_pie.setGeometry(QtCore.QRect(710, 240, 93, 28))

self.pushButton\_pie.setStyleSheet("background-color: rgb(109, 109, 109);")

self.pushButton\_pie.setObjectName("pushButton\_pie")

self.pushButton\_scat = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_scat.setGeometry(QtCore.QRect(840, 240, 93, 28))

self.pushButton\_scat.setStyleSheet("background-color: rgb(109, 109, 109);")

self.pushButton\_scat.setObjectName("pushButton\_scat")

self.pushButton\_about = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_about.setGeometry(QtCore.QRect(900, 160, 93, 28))

self.pushButton\_about.setStyleSheet("color: rgb(255, 255, 255);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(255, 255, 255, 255));")

self.pushButton\_about.setObjectName("pushButton\_about")

self.pushButton\_light = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_light.setGeometry(QtCore.QRect(790, 160, 93, 28))

self.pushButton\_light.setObjectName("pushButton\_light")

self.pushButton\_light.setStyleSheet("color: rgb(255, 255, 255);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(255, 255, 255, 255));")

self.pushButton\_light.setObjectName("pushButton\_light")

self.pushButton\_try = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_try.setGeometry(QtCore.QRect(680, 160, 93, 28))

self.pushButton\_try.setObjectName("pushButton\_try")

self.pushButton\_try.setStyleSheet("color: rgb(255, 255, 255);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(255, 255, 255, 255));")

self.pushButton\_try.setObjectName("pushButton\_try")

self.pushButton\_quiz = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_quiz.setGeometry(QtCore.QRect(570, 160, 93, 28))

self.pushButton\_quiz.setObjectName("pushButton\_quiz")

self.pushButton\_quiz.setStyleSheet("color: rgb(255, 255, 255);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(255, 255, 255, 255));")

self.pushButton\_quiz.setObjectName("pushButton\_quiz")

self.pushButton\_plot = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_plot.setGeometry(QtCore.QRect(460, 160, 93, 28))

self.pushButton\_plot.setObjectName("pushButton\_plot")

self.pushButton\_plot.setStyleSheet("color: rgb(255, 255, 255);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(255, 255, 255, 255));")

self.pushButton\_plot.setObjectName("pushButton\_plot")

self.photo = QtWidgets.QLabel(self.centralwidget)

self.photo.setGeometry(QtCore.QRect(14, 285, 971, 431))

self.photo.setText("")

self.photo.setPixmap(QtGui.QPixmap("test.jpg"))

self.photo.setScaledContents(True)

self.photo.setObjectName("photo")

MainWindow.setCentralWidget(self.centralwidget)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

self.pushButton\_quiz.clicked.connect(self.show\_pushButton\_quiz)

self.pushButton\_plot.clicked.connect(self.show\_pushButton\_plot)

self.pushButton\_try.clicked.connect(self.show\_pushButton\_try)

self.pushButton\_light.clicked.connect(self.show\_pushButton\_light)

self.pushButton\_about.clicked.connect(self.show\_pushButton\_about)

self.pushButton\_about.clicked.connect(self.show\_pushButton\_about)

self.pushButton\_bar.clicked.connect(self.show\_pushButton\_bar)

self.pushButton\_box.clicked.connect(self.show\_pushButton\_box)

self.pushButton\_freq.clicked.connect(self.show\_pushButton\_freq)

self.pushButton\_hist.clicked.connect(self.show\_pushButton\_hist)

self.pushButton\_line.clicked.connect(self.show\_pushButton\_line)

self.pushButton\_pie.clicked.connect(self.show\_pushButton\_pie)

self.pushButton\_scat.clicked.connect(self.show\_pushButton\_scat)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MyPY App"))

self.label\_title.setText(\_translate("MainWindow", "MyPY App"))

self.label\_tagline.setText(\_translate("MainWindow", "~ A Probono Platfrom"))

self.label\_hello.setText(\_translate("MainWindow", "Hello, Welcome back!"))

self.label\_learn.setText(\_translate("MainWindow", "What would you like to learn today?"))

self.pushButton\_bar.setText(\_translate("MainWindow", "Bar Graph"))

self.pushButton\_box.setText(\_translate("MainWindow", "Box Plot"))

self.pushButton\_freq.setText(\_translate("MainWindow", "Frequency Polygon"))

self.pushButton\_hist.setText(\_translate("MainWindow", "Histogram"))

self.pushButton\_line.setText(\_translate("MainWindow", "Line Graph"))

self.pushButton\_pie.setText(\_translate("MainWindow", "Pie Chart"))

self.pushButton\_scat.setText(\_translate("MainWindow", "Scatter Chart"))

self.pushButton\_about.setText(\_translate("MainWindow", "About"))

self.pushButton\_light.setText(\_translate("MainWindow", "Light Mode"))

self.pushButton\_try.setText(\_translate("MainWindow", "Try It Yourself"))

self.pushButton\_quiz.setText(\_translate("MainWindow", "Quiz"))

self.pushButton\_plot.setText(\_translate("MainWindow", "Plot Graphs"))

def show\_pushButton\_about(self):

self.photo.setPixmap(QtGui.QPixmap("res/misc/about\_d.jpg"))

def show\_pushButton\_bar(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/dark/bar.jpg"))

def show\_pushButton\_box(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/dark/box.jpg"))

def show\_pushButton\_freq(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/dark/freq.jpg"))

def show\_pushButton\_hist(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/dark/hist.jpg"))

def show\_pushButton\_line(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/dark/line.jpg"))

def show\_pushButton\_pie(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/dark/pie.jpg"))

def show\_pushButton\_scat(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/dark/scatter.jpg"))

def show\_pushButton\_light(self):

call(["python", "App\_L.py"])

def show\_pushButton\_try(self):

webbrowser.get(chrome\_path).open(url\_try)

def show\_pushButton\_quiz(self):

webbrowser.get(chrome\_path).open(url\_quiz)

def show\_pushButton\_plot(self):

webbrowser.get(chrome\_path).open(url\_plot)

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

from PyQt5 import QtCore, QtGui, QtWidgets

from subprocess import call

import webbrowser

url\_try = "https://colab.research.google.com/"

url\_plot = url = "http://localhost:8888/notebooks/Plot.ipynb"

url\_quiz = "http://localhost:8888/notebooks/Quiz.ipynb"

chrome\_path = 'C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s'

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(1000, 750)

palette = QtGui.QPalette()

brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)

brush = QtGui.QBrush(QtGui.QColor(109, 109, 109))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)

brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.WindowText, brush)

brush = QtGui.QBrush(QtGui.QColor(109, 109, 109))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)

brush = QtGui.QBrush(QtGui.QColor(120, 120, 120))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.WindowText, brush)

brush = QtGui.QBrush(QtGui.QColor(240, 240, 240))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)

MainWindow.setPalette(palette)

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setBold(True)

font.setWeight(75)

MainWindow.setFont(font)

icon = QtGui.QIcon()

icon.addPixmap(QtGui.QPixmap("res/ico/MyPY.ico"), QtGui.QIcon.Normal, QtGui.QIcon.Off)

MainWindow.setWindowIcon(icon)

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setStyleSheet("background-color: rgb(202, 202, 202);")

self.centralwidget.setObjectName("centralwidget")

self.label\_title = QtWidgets.QLabel(self.centralwidget)

self.label\_title.setGeometry(QtCore.QRect(340, 10, 301, 81))

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setPointSize(36)

font.setBold(True)

font.setWeight(75)

self.label\_title.setFont(font)

self.label\_title.setStyleSheet("color: rgb(35, 35, 35);")

self.label\_title.setObjectName("label\_title")

self.label\_tagline = QtWidgets.QLabel(self.centralwidget)

self.label\_tagline.setGeometry(QtCore.QRect(338, 100, 401, 31))

font = QtGui.QFont()

font.setFamily("Lucida Handwriting")

font.setPointSize(14)

self.label\_tagline.setFont(font)

self.label\_tagline.setStyleSheet("color: rgb(35, 35, 35);")

self.label\_tagline.setObjectName("label\_tagline")

self.line = QtWidgets.QFrame(self.centralwidget)

self.line.setGeometry(QtCore.QRect(7, 140, 991, 20))

self.line.setStyleSheet("color : rgb(35, 35, 35)")

self.line.setFrameShape(QtWidgets.QFrame.HLine)

self.line.setFrameShadow(QtWidgets.QFrame.Sunken)

self.line.setObjectName("line")

self.label\_hello = QtWidgets.QLabel(self.centralwidget)

self.label\_hello.setGeometry(QtCore.QRect(10, 160, 251, 31))

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setPointSize(14)

font.setBold(True)

font.setWeight(75)

self.label\_hello.setFont(font)

self.label\_hello.setStyleSheet("color: rgb(35, 35, 35);")

self.label\_hello.setObjectName("label\_hello")

self.label\_learn = QtWidgets.QLabel(self.centralwidget)

self.label\_learn.setGeometry(QtCore.QRect(10, 200, 291, 21))

font = QtGui.QFont()

font.setFamily("Century Gothic")

font.setPointSize(10)

font.setBold(True)

font.setWeight(75)

self.label\_learn.setFont(font)

self.label\_learn.setStyleSheet("color: rgb(35, 35, 35);")

self.label\_learn.setObjectName("label\_learn")

self.pushButton\_bar = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_bar.setGeometry(QtCore.QRect(10, 240, 93, 28))

self.pushButton\_bar.setStyleSheet("color: rgb(35, 35, 35);\n"

"")

self.pushButton\_bar.setObjectName("pushButton\_bar")

self.pushButton\_box = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_box.setGeometry(QtCore.QRect(140, 240, 93, 28))

self.pushButton\_box.setStyleSheet("color: rgb(35, 35, 35);")

self.pushButton\_box.setObjectName("pushButton\_box")

self.pushButton\_freq = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_freq.setGeometry(QtCore.QRect(270, 240, 141, 28))

self.pushButton\_freq.setStyleSheet("color: rgb(35, 35, 35);")

self.pushButton\_freq.setObjectName("pushButton\_freq")

self.pushButton\_hist = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_hist.setGeometry(QtCore.QRect(450, 240, 93, 28))

self.pushButton\_hist.setStyleSheet("color: rgb(35, 35, 35);")

self.pushButton\_hist.setObjectName("pushButton\_hist")

self.pushButton\_line = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_line.setGeometry(QtCore.QRect(580, 240, 93, 28))

self.pushButton\_line.setStyleSheet("color: rgb(35, 35, 35);")

self.pushButton\_line.setObjectName("pushButton\_line")

self.pushButton\_pie = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_pie.setGeometry(QtCore.QRect(710, 240, 93, 28))

self.pushButton\_pie.setStyleSheet("color: rgb(35, 35, 35);")

self.pushButton\_pie.setObjectName("pushButton\_pie")

self.pushButton\_scat = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_scat.setGeometry(QtCore.QRect(840, 240, 93, 28))

self.pushButton\_scat.setStyleSheet("color: rgb(35, 35, 35);")

self.pushButton\_scat.setObjectName("pushButton\_scat")

self.pushButton\_about = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_about.setGeometry(QtCore.QRect(900, 160, 93, 28))

self.pushButton\_about.setStyleSheet("color: rgb(35, 35, 35);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(35, 0, 0, 35));")

self.pushButton\_about.setObjectName("pushButton\_about")

self.pushButton\_dark = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_dark.setGeometry(QtCore.QRect(790, 160, 93, 28))

self.pushButton\_dark.setObjectName("pushButton\_dark")

self.pushButton\_dark.setStyleSheet("color: rgb(35, 35, 35);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(35, 0, 0, 35));")

self.pushButton\_dark.setObjectName("pushButton\_dark")

self.pushButton\_try = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_try.setGeometry(QtCore.QRect(680, 160, 93, 28))

self.pushButton\_try.setObjectName("pushButton\_try")

self.pushButton\_try.setStyleSheet("color: rgb(35, 35, 35);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(225, 0, 0, 225));")

self.pushButton\_try.setObjectName("pushButton\_try")

self.pushButton\_quiz = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_quiz.setGeometry(QtCore.QRect(570, 160, 93, 28))

self.pushButton\_quiz.setObjectName("pushButton\_quiz")

self.pushButton\_quiz.setStyleSheet("color: rgb(35, 35, 35);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(225, 0, 0, 225));")

self.pushButton\_quiz.setObjectName("pushButton\_quiz")

self.pushButton\_plot = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_plot.setGeometry(QtCore.QRect(460, 160, 93, 28))

self.pushButton\_plot.setObjectName("pushButton\_plot")

self.pushButton\_plot.setStyleSheet("color: rgb(35, 35, 35);\n"

"color: qlineargradient(spread:pad, x1:0, y1:0, x2:1, y2:0, stop:0 rgba(255, 0, 0, 255), stop:1 rgba(225, 0, 0, 225));")

self.pushButton\_plot.setObjectName("pushButton\_plot")

self.photo = QtWidgets.QLabel(self.centralwidget)

self.photo.setGeometry(QtCore.QRect(14, 285, 971, 431))

self.photo.setText("")

self.photo.setPixmap(QtGui.QPixmap("test.jpg"))

self.photo.setScaledContents(True)

self.photo.setObjectName("photo")

MainWindow.setCentralWidget(self.centralwidget)

self.statusbar = QtWidgets.QStatusBar(MainWindow)

self.statusbar.setObjectName("statusbar")

MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

self.pushButton\_quiz.clicked.connect(self.show\_pushButton\_quiz)

self.pushButton\_plot.clicked.connect(self.show\_pushButton\_plot)

self.pushButton\_dark.clicked.connect(self.show\_pushButton\_dark)

self.pushButton\_about.clicked.connect(self.show\_pushButton\_about)

self.pushButton\_bar.clicked.connect(self.show\_pushButton\_bar)

self.pushButton\_box.clicked.connect(self.show\_pushButton\_box)

self.pushButton\_freq.clicked.connect(self.show\_pushButton\_freq)

self.pushButton\_hist.clicked.connect(self.show\_pushButton\_hist)

self.pushButton\_line.clicked.connect(self.show\_pushButton\_line)

self.pushButton\_pie.clicked.connect(self.show\_pushButton\_pie)

self.pushButton\_scat.clicked.connect(self.show\_pushButton\_scat)

self.pushButton\_try.clicked.connect(self.show\_pushButton\_try)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "MyPY App"))

self.label\_title.setText(\_translate("MainWindow", "MyPY App"))

self.label\_tagline.setText(\_translate("MainWindow", "~ A Probono Platfrom"))

self.label\_hello.setText(\_translate("MainWindow", "Hello, Welcome back!"))

self.label\_learn.setText(\_translate("MainWindow", "What would you like to learn today?"))

self.pushButton\_bar.setText(\_translate("MainWindow", "Bar Graph"))

self.pushButton\_box.setText(\_translate("MainWindow", "Box Plot"))

self.pushButton\_freq.setText(\_translate("MainWindow", "Frequency Polygon"))

self.pushButton\_hist.setText(\_translate("MainWindow", "Histogram"))

self.pushButton\_line.setText(\_translate("MainWindow", "Line Graph"))

self.pushButton\_pie.setText(\_translate("MainWindow", "Pie Chart"))

self.pushButton\_scat.setText(\_translate("MainWindow", "Scatter Chart"))

self.pushButton\_about.setText(\_translate("MainWindow", "About"))

self.pushButton\_dark.setText(\_translate("MainWindow", "Dark Mode"))

self.pushButton\_try.setText(\_translate("MainWindow", "Try It Yourself"))

self.pushButton\_quiz.setText(\_translate("MainWindow", "Quiz"))

self.pushButton\_plot.setText(\_translate("MainWindow", "Plot Graphs"))

def show\_pushButton\_about(self):

self.photo.setPixmap(QtGui.QPixmap("res/misc/about\_l.jpg"))

def show\_pushButton\_bar(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/light/bar.jpg"))

def show\_pushButton\_box(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/light/box.jpg"))

def show\_pushButton\_freq(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/light/freq.jpg"))

def show\_pushButton\_hist(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/light/hist.jpg"))

def show\_pushButton\_line(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/light/line.jpg"))

def show\_pushButton\_pie(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/light/pie.jpg"))

def show\_pushButton\_scat(self):

self.photo.setPixmap(QtGui.QPixmap("res/img/light/scatter.jpg"))

def show\_pushButton\_dark(self):

call(["python", "App\_D.py"])

def show\_pushButton\_try(self):

webbrowser.get(chrome\_path).open(url\_try)

def show\_pushButton\_quiz(self):

webbrowser.get(chrome\_path).open(url\_quiz)

def show\_pushButton\_plot(self):

webbrowser.get(chrome\_path).open(url\_plot)

if \_\_name\_\_ == "\_\_main\_\_":

import sys

app = QtWidgets.QApplication(sys.argv)

MainWindow = QtWidgets.QMainWindow()

ui = Ui\_MainWindow()

ui.setupUi(MainWindow)

MainWindow.show()

sys.exit(app.exec\_())

#PLOT PROGRAM

import matplotlib.pyplot as pl

import numpy as np

print("What kind of graph would you like to plot?")

print("Enter 1 for Line Graph")

print("Enter 2 for Bar Graph")

print("Enter 3 for Pie Chart")

print("Enter 4 for Histogram")

print("Enter 5 for Scatter Chart")

resp = int(input("Enter your choice: "))

print("Enter your values for the graph")

X = np.arange(5)

Y = eval(input("Enter any five values for X-Axis: "))

Z = eval(input("Enter any five values for Y-Axis:"))

if resp == 1:

pl.xlabel("X-Axis")

pl.ylabel("Y-Axis")

pl.title("Line Chart")

pl.plot(X, Y, marker = 'o', markeredgecolor = 'red', label = 'Line A', linewidth = 3, linestyle = 'dashdot')

pl.plot(X, Z, marker = 'o', markeredgecolor = 'green', label = 'Line B', linewidth = 3, linestyle = 'dashed')

pl.legend()

pl.grid("True", color = 'yellow')

pl.show()

elif resp == 2:

pl.bar(Y, Z, width = [0.6,0.7,0.8,0.9,1], color = ('red','blue','green','purple','orange'))

pl.xlabel('X-Axis')

pl.ylabel('Y-Axis')

pl.title('Bar Graph')

pl.show(Y,Z)

elif resp == 3:

exp = [0,0.2,0.3,0,0.1]

pl.pie(Y, colors = ['magenta','cyan','green','red','blue'], startangle = 90, explode = exp, shadow = True, autopct = '%.1f%%')

pl.title('Pie Chart')

pl.show()

elif resp == 4:

pl.hist([Y,Z],bins = [20,25,30,35], color = ['green','orange'], rwidth = 0.95, histtype = "barstacked", orientation = "horizontal")

pl.xlabel("Y-Axis")

pl.ylabel("X-Axis")

pl.title("Histogram")

elif resp == 5:

pl.scatter(Y, Z, c = ['r','g','b','purple','cyan'], s = 20)

pl.xlabel("X-Axis")

pl.ylabel("Y-Axis")

pl.title("Scatter Chart")

pl.show(Y,Z)

elif resp > 5:

print("Invalid input")

#QUIZ PROGRAM

import time

ans1 = 'A'

ans2 = 'B'

ans3 = 'C'

ans4 = 'A'

ans5 = 'B'

ans6 = 'C'

ans7 = 'B'

ans8 = 'C'

ans9 = 'C'

ans10 = 'B'

print("Hi, Let's test your knowledge!")

print("Please enter your choice of option as A/B/C/D as per the question")

print("Q1) Which of the following is a Python package used for 2D graphics? \n A) matplotlib.pyplot \n B) matplotlib.pip \n C) matplotlib.numpy \n D) matplotlib.plt")

x= input ("Enter the option: ")

sum=0

if x== ans1:

sum=sum+1

print('Next question is loading...')

time.sleep(1.5)

print('\n\t')

print("Q2) Identify the package manager for Python packages or modules \n A) Matplotlib \n B) PIP \n C) plt.show() \n D) python package")

x = input ("Enter the option: ")

if x== ans2:

sum=sum+1

print('Next question is loading...')

time.sleep(1.5)

print('\n\t')

print("Q3) Read the given below statements & identify the right options from the following for Pie Chart \n Statement A: To make a Pie Chart with Matplotlib, we can use the plt.pie() function. \n Statement B: The autopct parameter allows us to display the % value using the Python string formatting. \n A) Statement A is correct \n B) Statement B is correct\n C) Both the statements are correct\n D) Both the statements are incorrect")

x= input ("Enter the option: ")

if x== ans3:

sum=sum+1

print("Next question is loading...")

time.sleep(1.5)

print('\n\t')

print("Q4) Which of the following method will be used to display the plot? \n A) show() \n B) display() \n C) execute() \n D) plot()")

x= input ("Enter the option: ")

if x== ans4:

sum=sum+1

print("Next question is loading...")

time.sleep(1.5)

print('\n\t')

print('Q5) Which type of chart shows the relationship between a numerical variable and categorical variable? \n A) Line Chart \n B) Bar Chart \n C) Pie Chart \n D) XY plot')

x= input ("Enter the option: ")

if x== ans5:

sum=sum+1

print('Next question is loading...')

time.sleep(1.5)

print('\n\t')

print('Q6) Which plot displays the distribution of data based on the five number summary? \n A) Scatter Plot \n B) Line Plot \n C) Box Plot \n D)Chart Plot')

x= input ("Enter the option: ")

if x== ans6:

sum=sum+1

print('Next question is loading...')

time.sleep(1.5)

print('\n\t')

print("Q7) To give a title to X-Axis, which of the following method is useful? \n A) pl.xtitle(“title”) \n B) pl.xlabel(“title”) \n C) pl.xheader(“title”) \n D) pl.xlabel.show(“title”)")

x= input ("Enter the option: ")

if x== ans6:

sum=sum+1

print('Next question is loading...')

time.sleep(1.5)

print('\n\t')

print("Q8) To change the width of bars in Bar Chart, which of the following argument (float value) is used? \n A) thick \n B) thickness \n C) width \n D) barwidth")

x= input ("Enter the option: ")

if x== ans6:

sum=sum+1

print('Next question is loading...')

time.sleep(1.5)

print('\n\t')

print("Q9) Which method is used to display or show the legends of a plot? \n A) pl.show() \n B) pl.display() \n C) pl.legend() \n D) pl.values()")

x= input ("Enter the option: ")

if x== ans6:

sum=sum+1

print('Next question is loading...')

time.sleep(1.5)

print('\n\t')

print("Q10) What kind of plot(s) would you use to examine the distribution of a numeric variable? \n A) Scatter plot \n B) Boxplot \n C) Bar Plot \n D) Histogram")

x= input ("Enter the option: ")

if x== ans6:

sum=sum+1

time.sleep(1.5)

print('\n\t')

print('The quiz is over')

print('Please wait while your scores are being calculated....')

time.sleep(2.5)

if sum == 10:

print('Your score is: \n', sum,'/10')

print("Keep up the good work!")

elif sum < 3:

print('Your score is: \n', sum,'/10')

print("Mistakes happen, Revise your concepts and attempt the questions carefully!")

elif sum < 6:

print('Your score is: \n', sum,'/10')

print("Don't feel bad, there is still room for improvement!")

elif sum < 9:

print('Your score is: \n', sum,'/10')

print("Almost there, you just need some more practise!")

else:

print('Your score is: \n', sum,'/10')