



University of
Dayton

Data Visualization with Python (Lab 9)

Data Visualization

Dr. Tam Nguyen

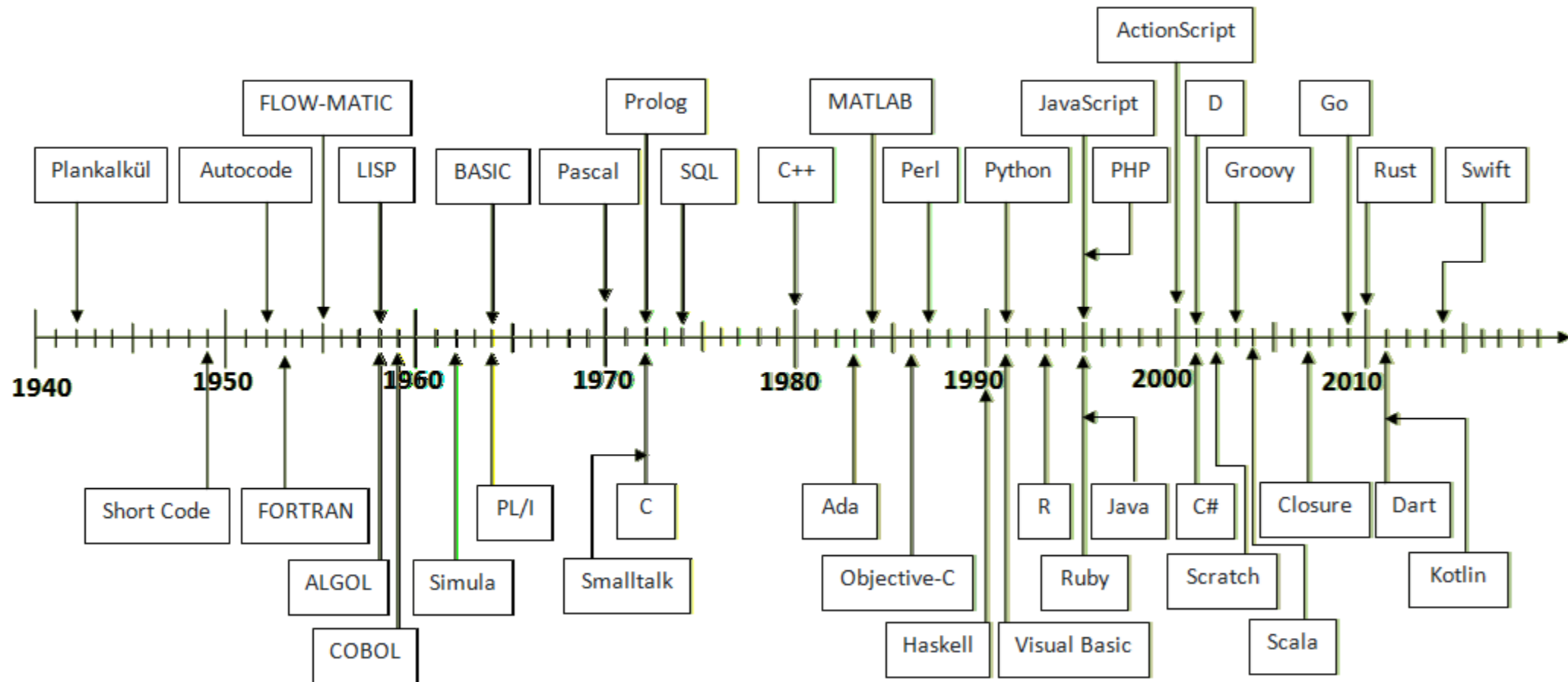
tamnguyen@udayton.edu

Objective

- Using Python for Data Visualization
 - Install Python IDE
 - Use Python to plot charts/diagrams with the given data



Python in history



Matplotlib

- Matplotlib is a visualization library in Python for 2D plots of arrays.
- In particular, Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

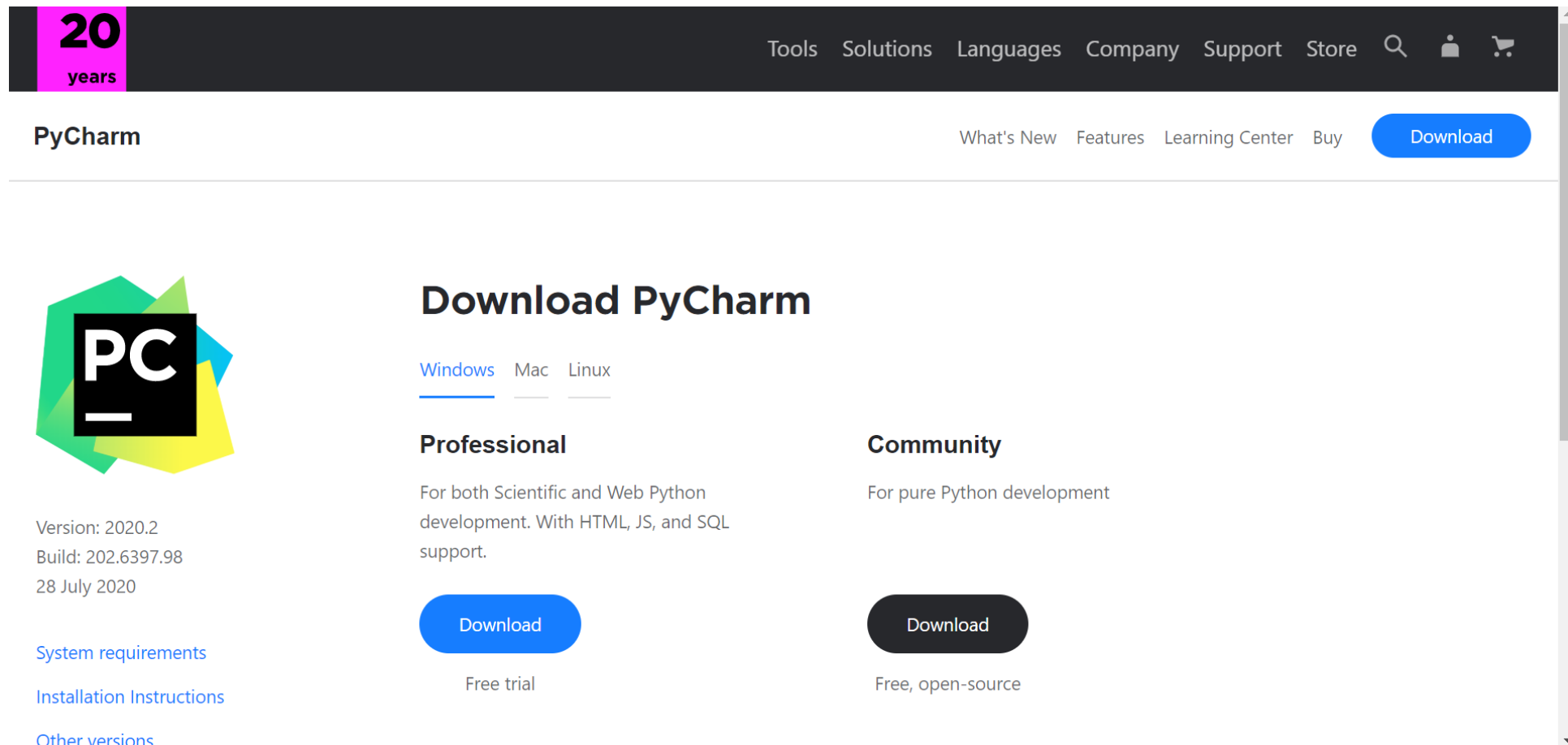


What can be drawn with Matplotlib?

- Matplotlib comes with a wide variety of plots
 - Basic graphs: bar graph, line chart, histogram, scatter plot
 - Advanced graphs: treemap, parallel coordinates.

Install PyCharm IDE

- <https://www.jetbrains.com/pycharm/download/#section=windows>



The screenshot shows the JetBrains PyCharm download page. At the top, there's a dark navigation bar with a '20 years' badge on the left and links for Tools, Solutions, Languages, Company, Support, and Store on the right. Below this, the 'PyCharm' header is on the left, and links for 'What's New', 'Features', 'Learning Center', and 'Buy' are on the right, followed by a blue 'Download' button. The main content area is divided into two columns. The left column features the PyCharm logo (a green and yellow hexagon with 'PC' and a minus sign) and version information: 'Version: 2020.2', 'Build: 202.6397.98', and '28 July 2020'. Below this are links for 'System requirements', 'Installation Instructions', and 'Other versions'. The right column is titled 'Download PyCharm' and has tabs for 'Windows' (selected), 'Mac', and 'Linux'. Under the 'Windows' tab, there are two sections: 'Professional' and 'Community'. The 'Professional' section describes it as 'For both Scientific and Web Python development. With HTML, JS, and SQL support.' and includes a blue 'Download' button with the text 'Free trial' below it. The 'Community' section describes it as 'For pure Python development' and includes a dark grey 'Download' button with the text 'Free, open-source' below it.

20 years

Tools Solutions Languages Company Support Store

PyCharm

What's New Features Learning Center Buy [Download](#)

Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)

Free trial

Community

For pure Python development

[Download](#)

Free, open-source

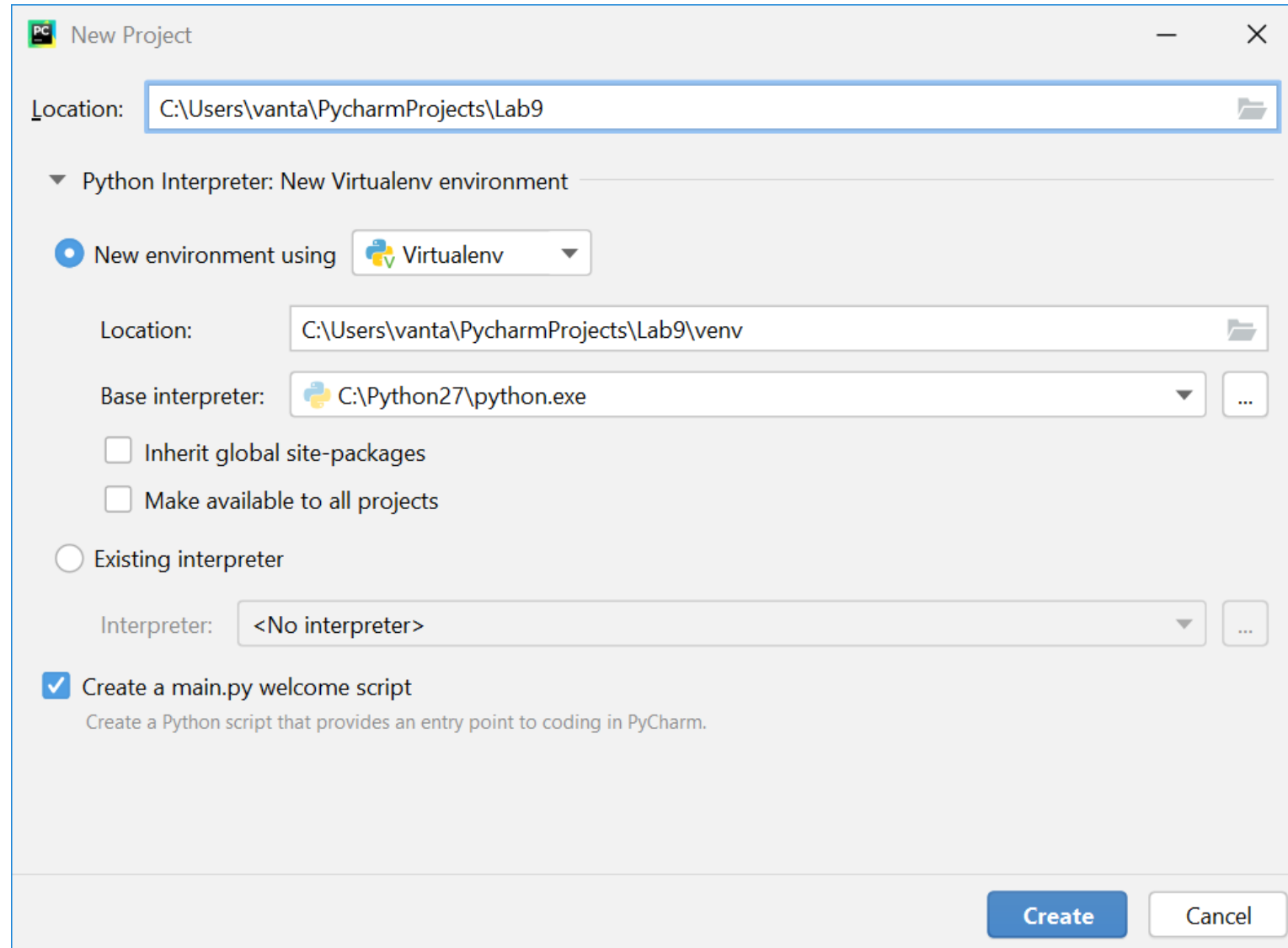
Version: 2020.2
Build: 202.6397.98
28 July 2020

[System requirements](#)
[Installation Instructions](#)
[Other versions](#)

If PyCharm was not installed

- Visit Online Python:
- <https://repl.it/languages/python3>

Create Lab9 project



The image shows the 'New Project' dialog box in PyCharm. The 'Location' field is set to 'C:\Users\vanta\PycharmProjects\Lab9'. Under the 'Python Interpreter' section, the 'New Virtualenv environment' option is selected. The 'New environment using' dropdown is set to 'Virtualenv'. The 'Location' for the virtual environment is 'C:\Users\vanta\PycharmProjects\Lab9\venv'. The 'Base interpreter' is 'C:\Python27\python.exe'. The 'Inherit global site-packages' and 'Make available to all projects' checkboxes are unchecked. The 'Existing interpreter' option is not selected. The 'Interpreter' dropdown is set to '<No interpreter>'. The 'Create a main.py welcome script' checkbox is checked, with a description below it: 'Create a Python script that provides an entry point to coding in PyCharm.' At the bottom right, there are 'Create' and 'Cancel' buttons.

PC New Project

Location: C:\Users\vanta\PycharmProjects\Lab9

Python Interpreter: New Virtualenv environment

☒ New environment using Virtualenv

Location: C:\Users\vanta\PycharmProjects\Lab9\venv

Base interpreter: C:\Python27\python.exe

☐ Inherit global site-packages

☐ Make available to all projects

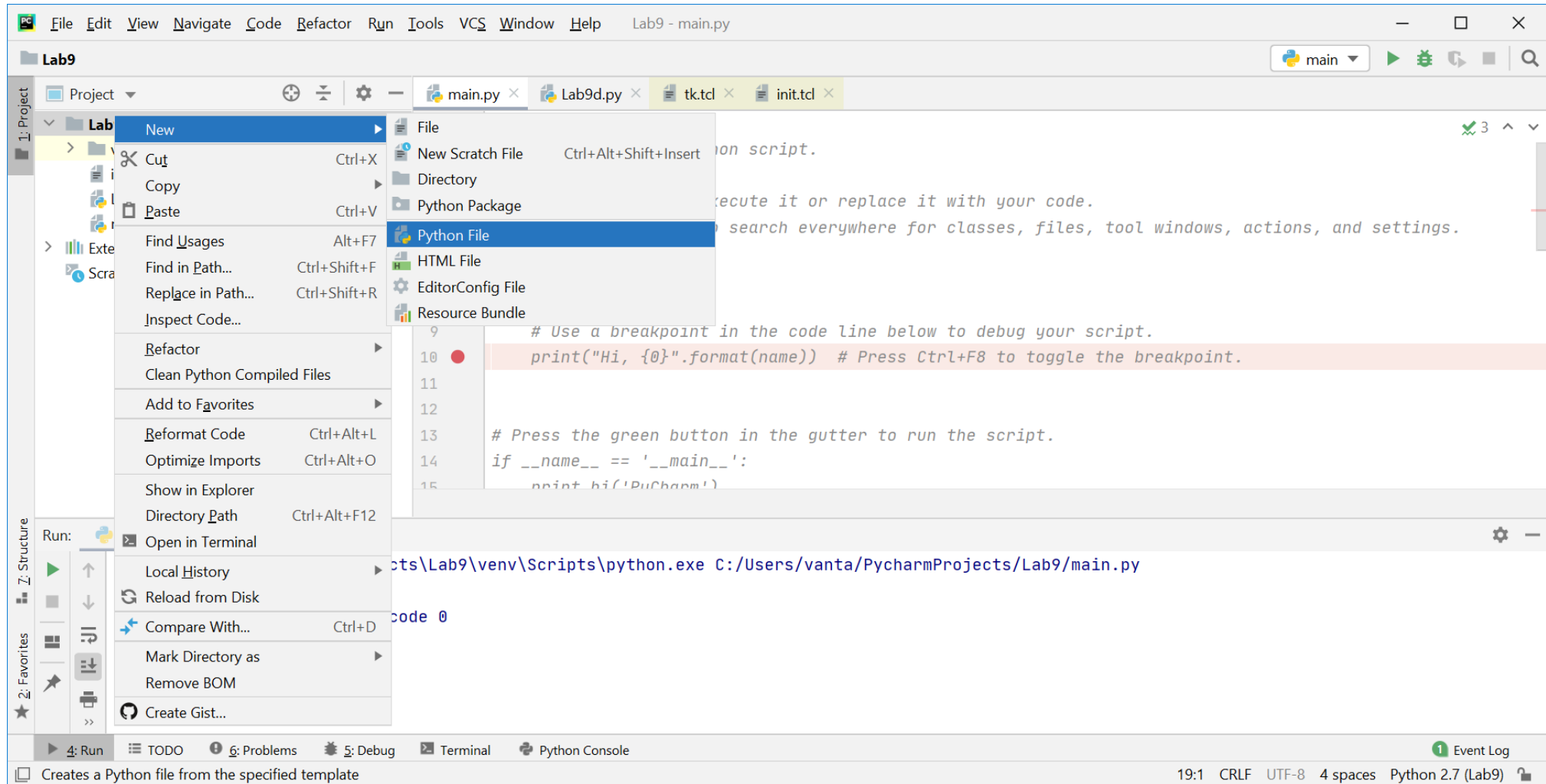
☐ Existing interpreter

Interpreter: <No interpreter>

☒ Create a main.py welcome script
Create a Python script that provides an entry point to coding in PyCharm.

Create Cancel

Create Python file: main.py



Hello World



Create a vector and display it



The image shows a PyCharm IDE window with a Python file named `main.py`. The code in the editor is:

```
1 print('Hello World')
2 a = [1,2,3,4]
3 print(a)
```

The code is executed, and the output is displayed in the Run console at the bottom. The output shows the command executed, the printed text "Hello World", the printed list `[1, 2, 3, 4]`, and a message indicating the process finished with exit code 0.

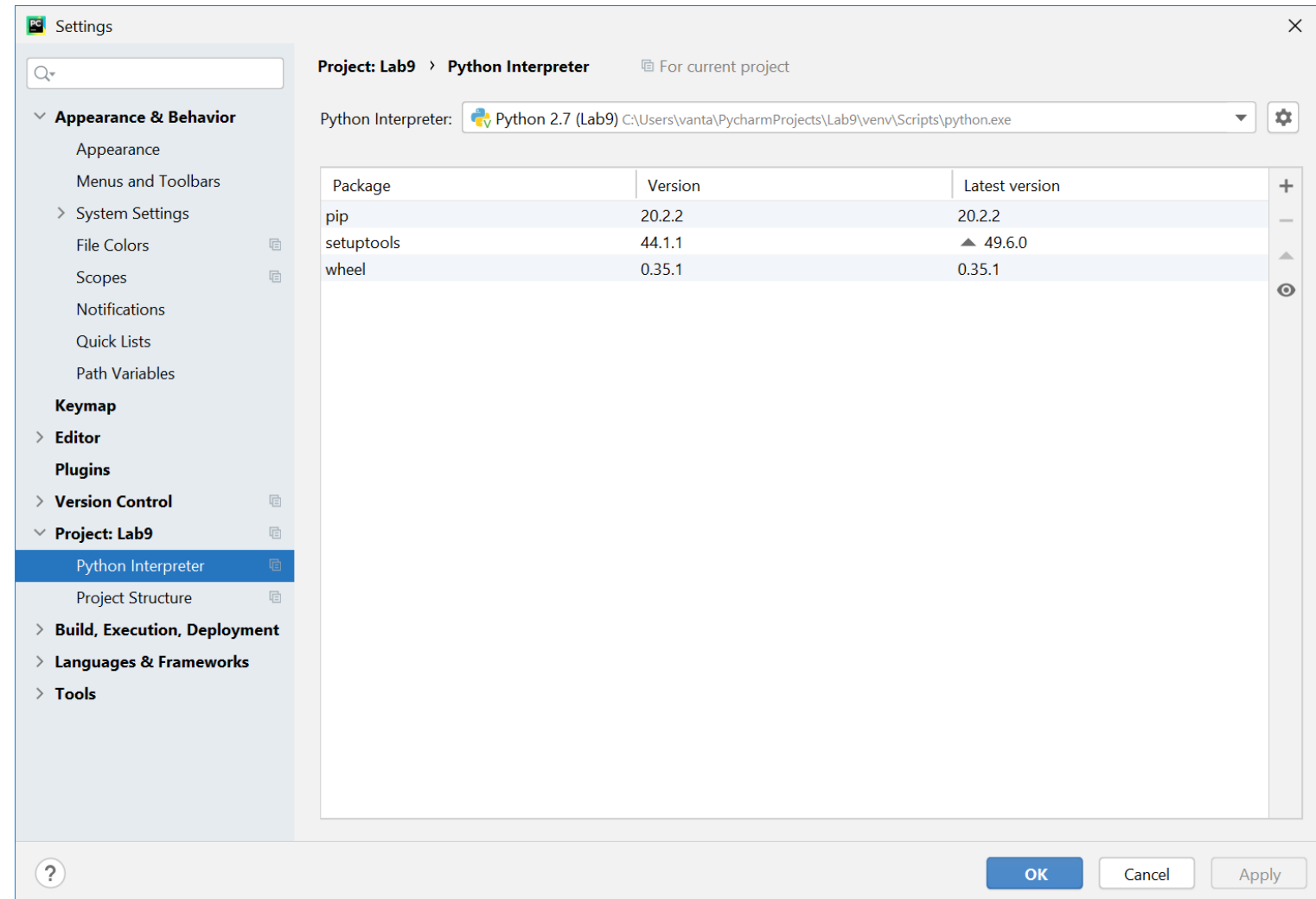
The Run console output is:

```
Run: C:\Users\vanta\PycharmProjects\Lab9\venv\Scripts\python.exe C:/Users/vanta/PycharmProjects/Lab9/main.py
Hello World
[1, 2, 3, 4]
Process finished with exit code 0
```

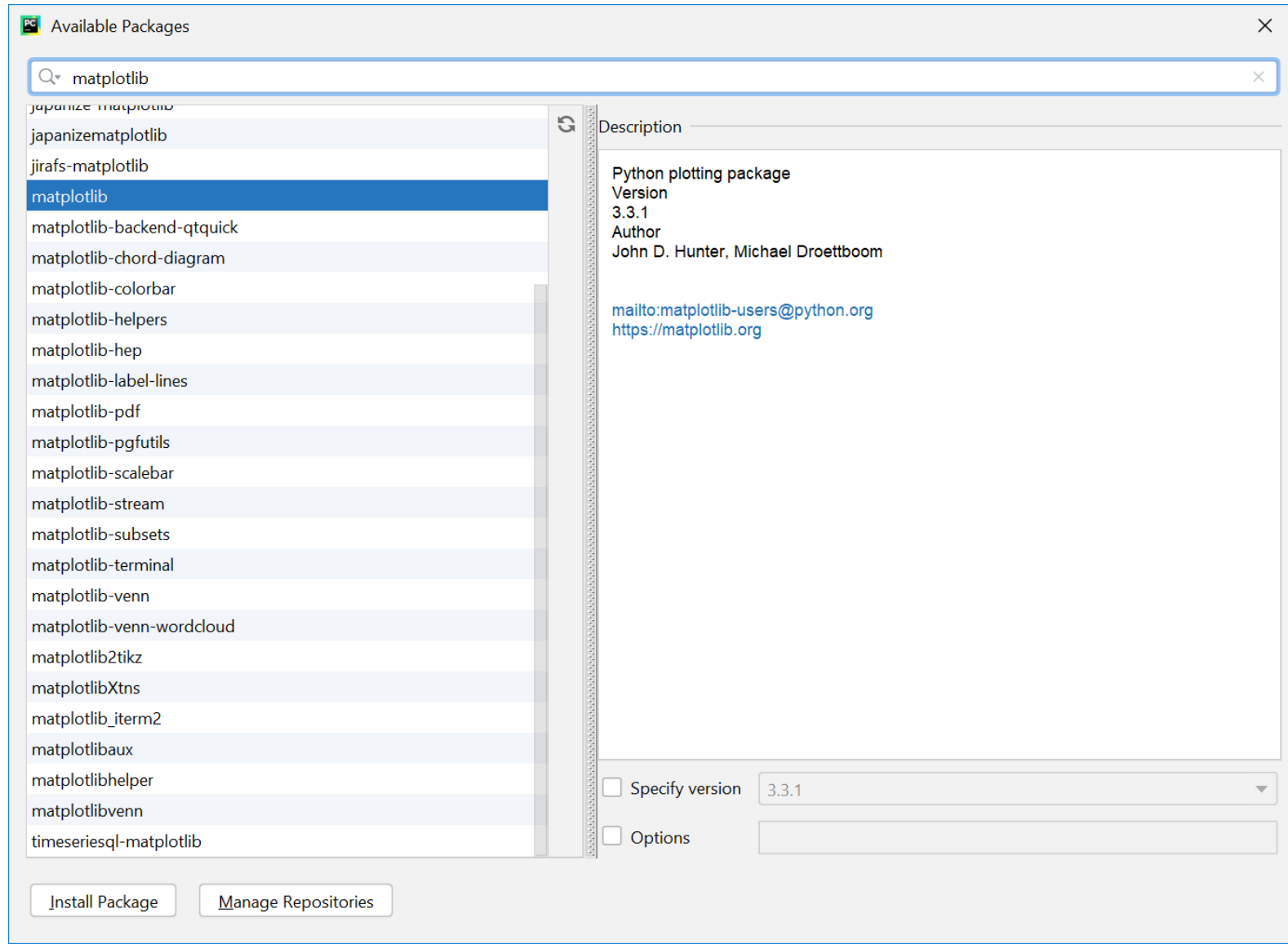
A red rectangle highlights the Run button (a green play icon) in the top right corner of the IDE window.

Update Settings

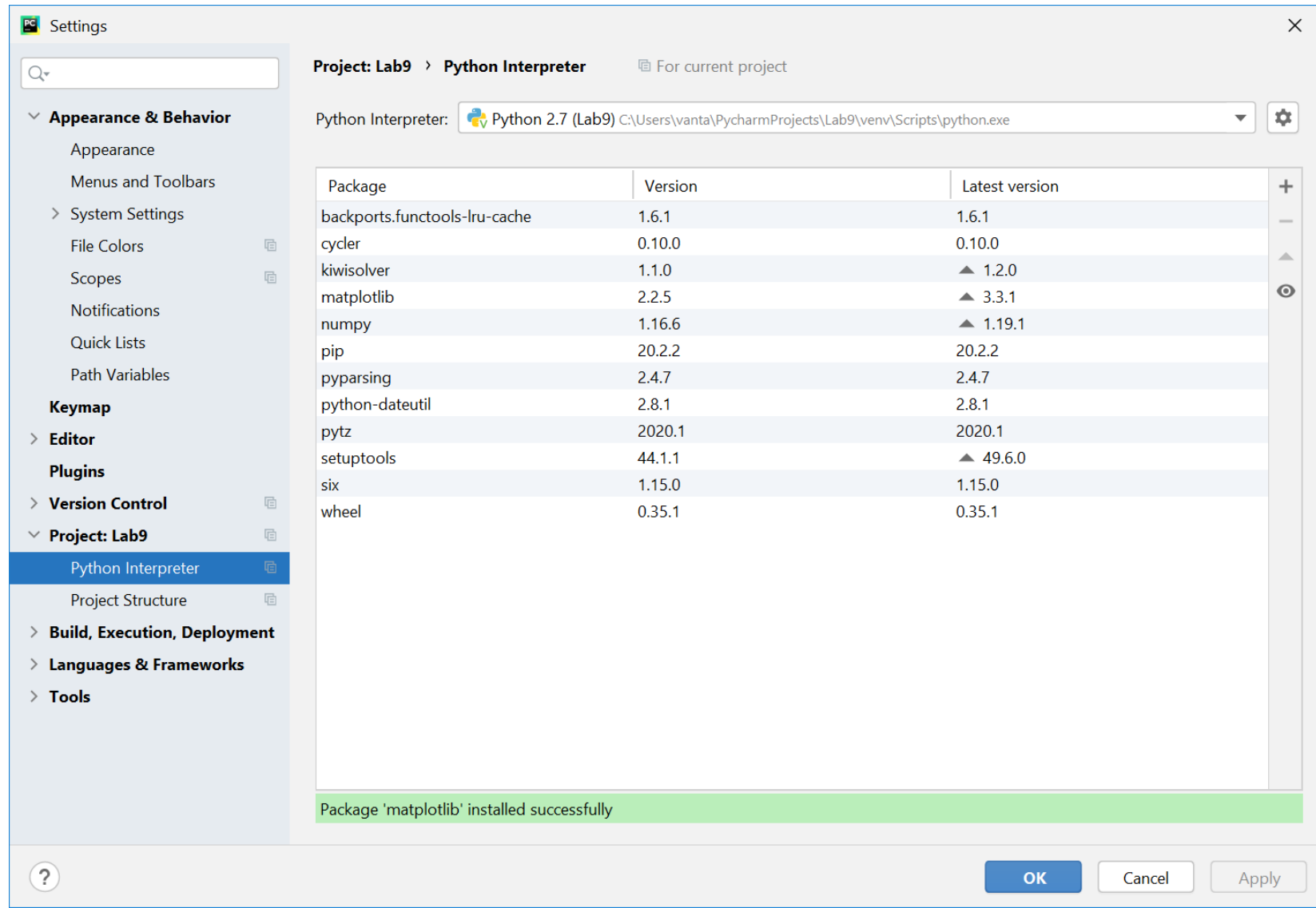
- File\Settings\Project:Lab9



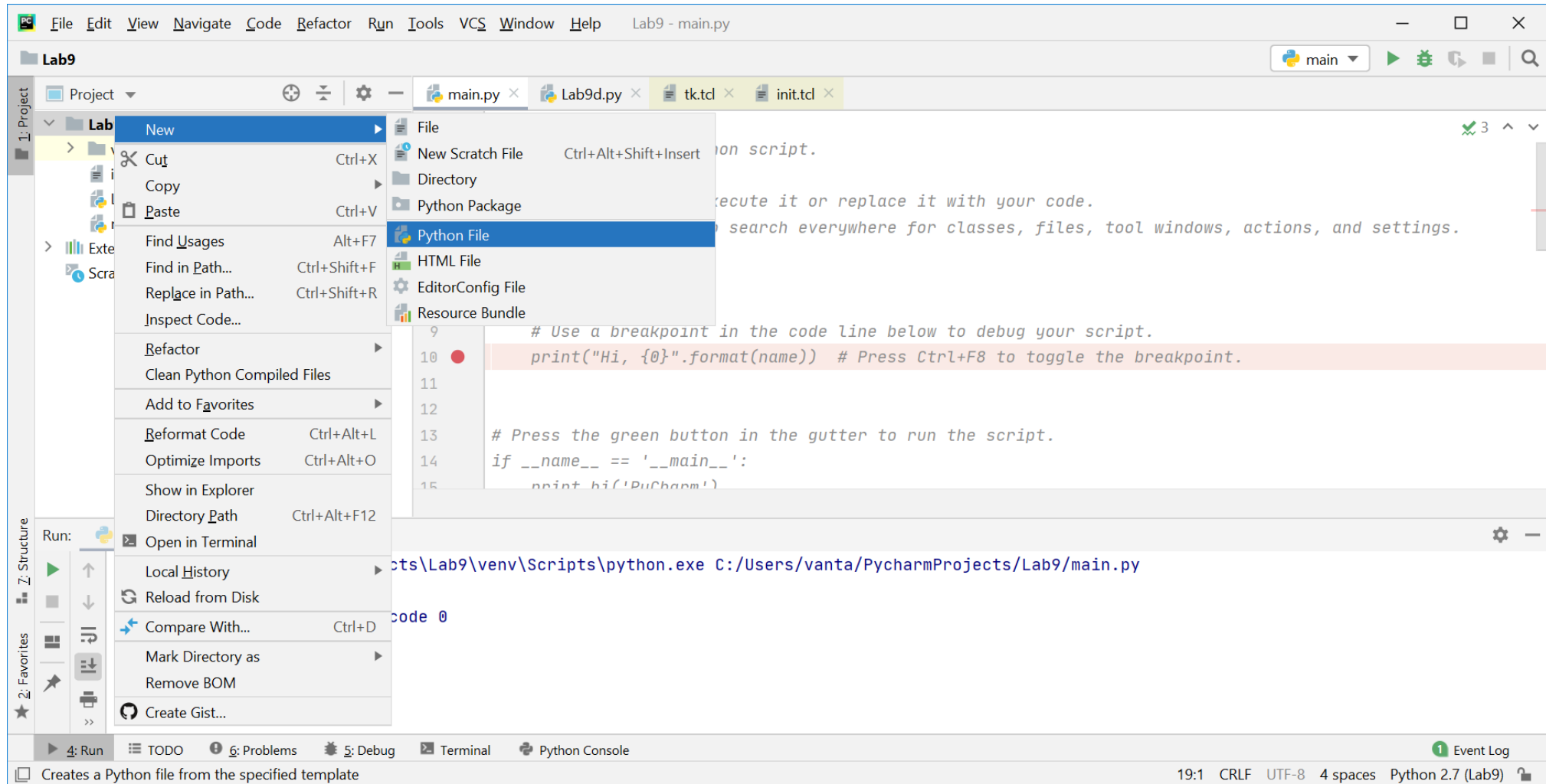
Add matplotlib



Installation finished



Create Python file: Lab9a.py



Lab9a.py

```
# importing matplotlib module
from matplotlib import pyplot as plt
def Lab9a():
    # x-axis values
    x = [5, 2, 9, 4, 7]

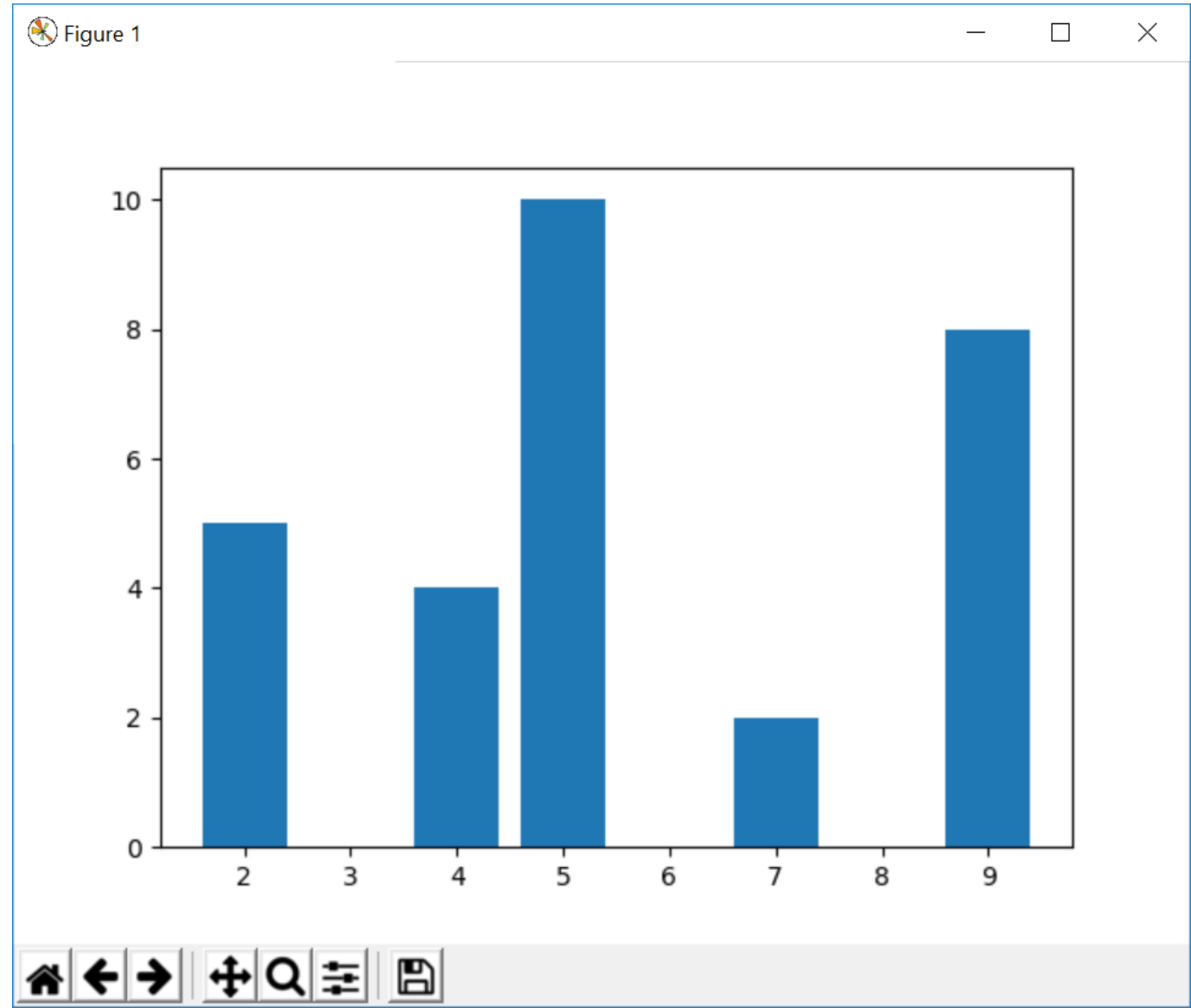
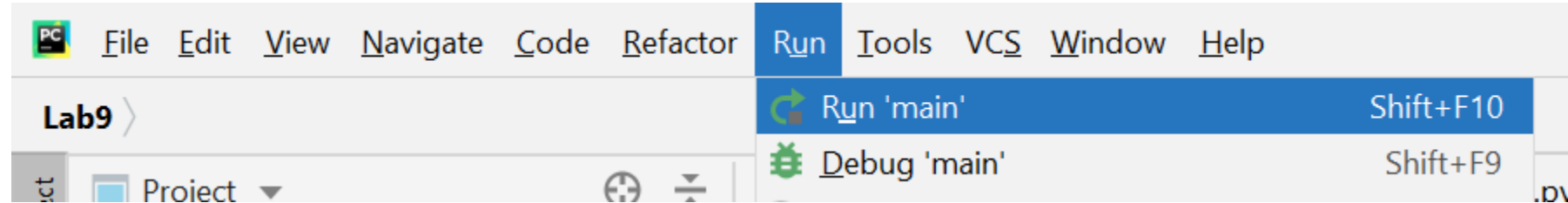
    # Y-axis values
    y = [10, 5, 8, 4, 2]

    # Create a new figure
    plt.figure(1)

    # Function to plot the bar
    plt.bar(x,y)
    plt.show()
```


main.py

```
from Lab9a import Lab9a  
Lab9a()
```

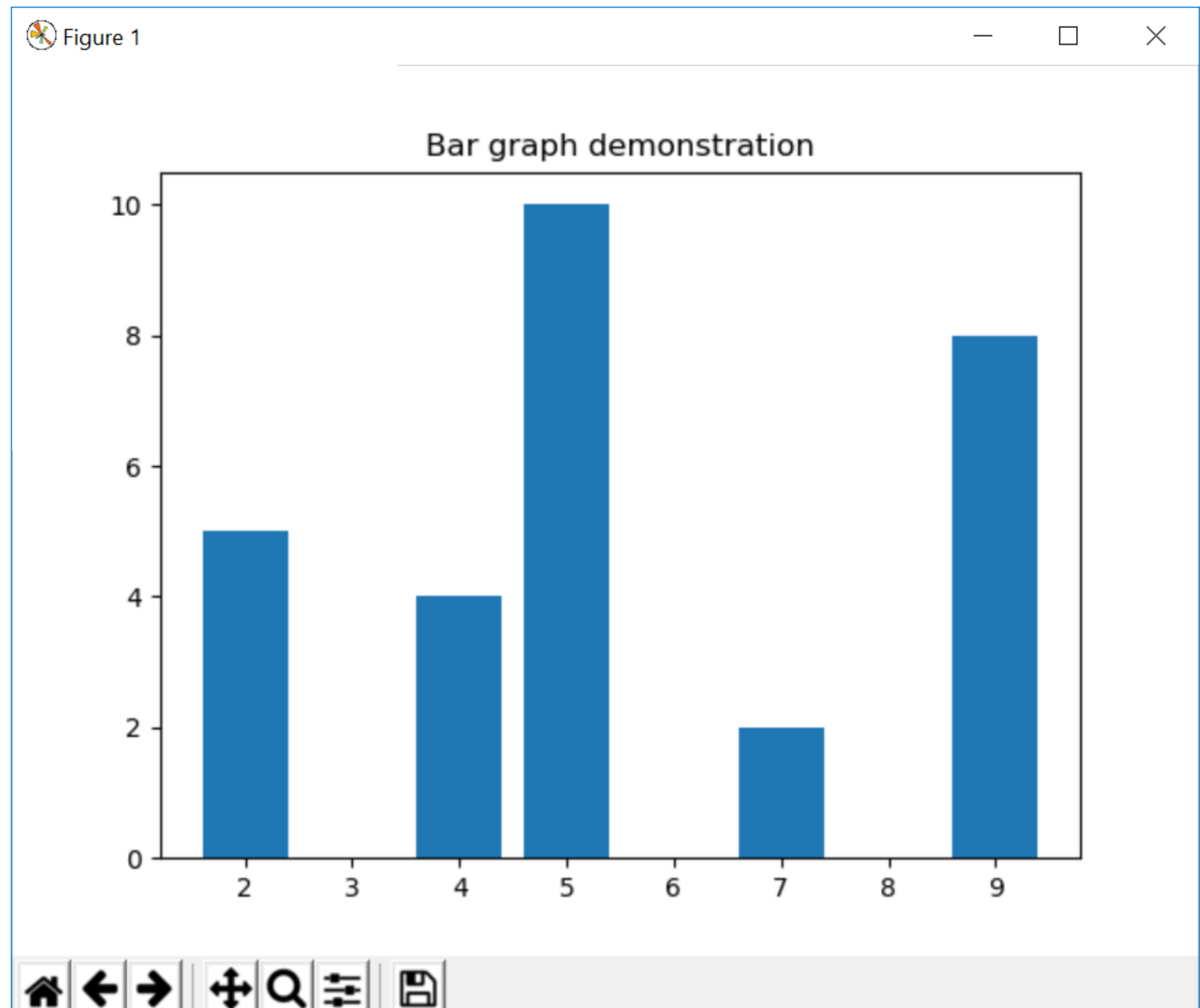
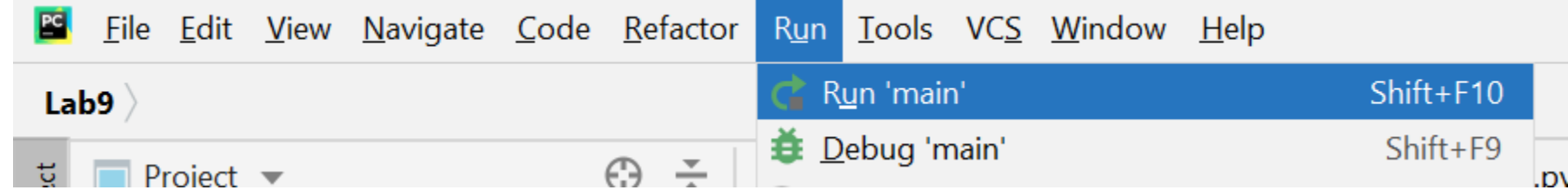


Lab9a.py

```
# importing matplotlib module
from matplotlib import pyplot as plt
def Lab9a():
    # x-axis values
    x = [5, 2, 9, 4, 7]
    # Y-axis values
    y = [10, 5, 8, 4, 2]
    # Create a new figure
    plt.figure(1)
    # Function to plot the bar
    plt.bar(x,y)
    plt.title("Bar graph demonstration")
    plt.show()
```

main.py

```
from Lab9a import Lab9a  
Lab9a()
```



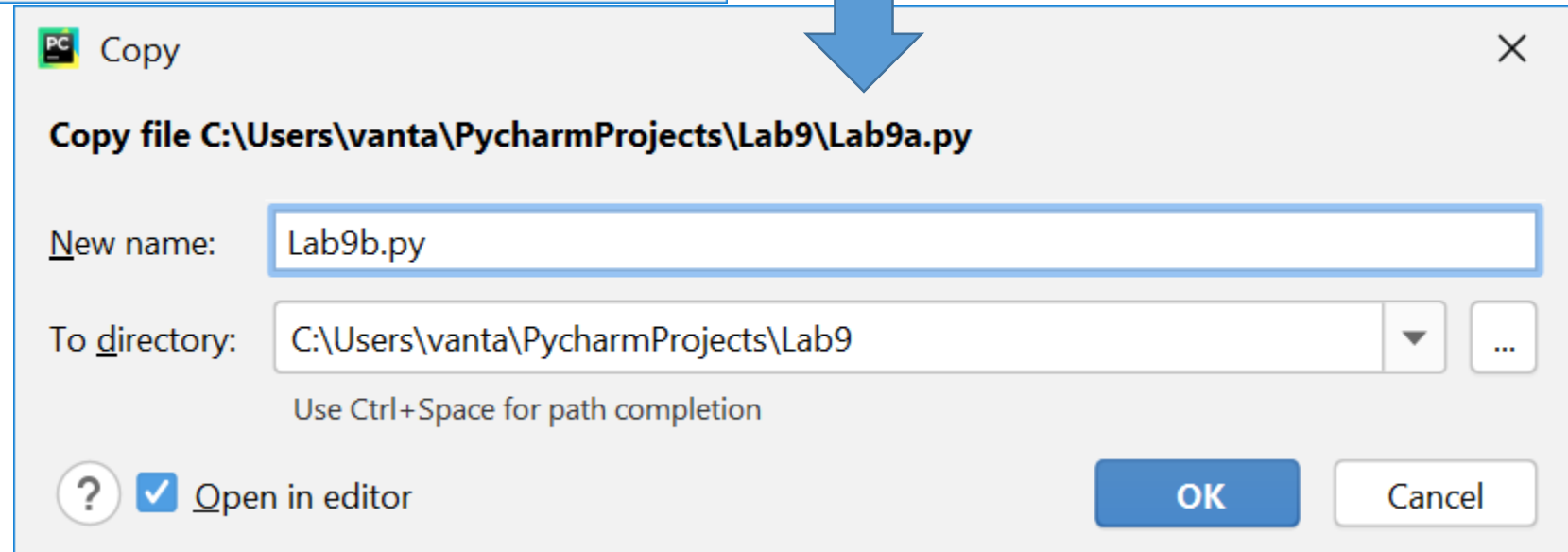
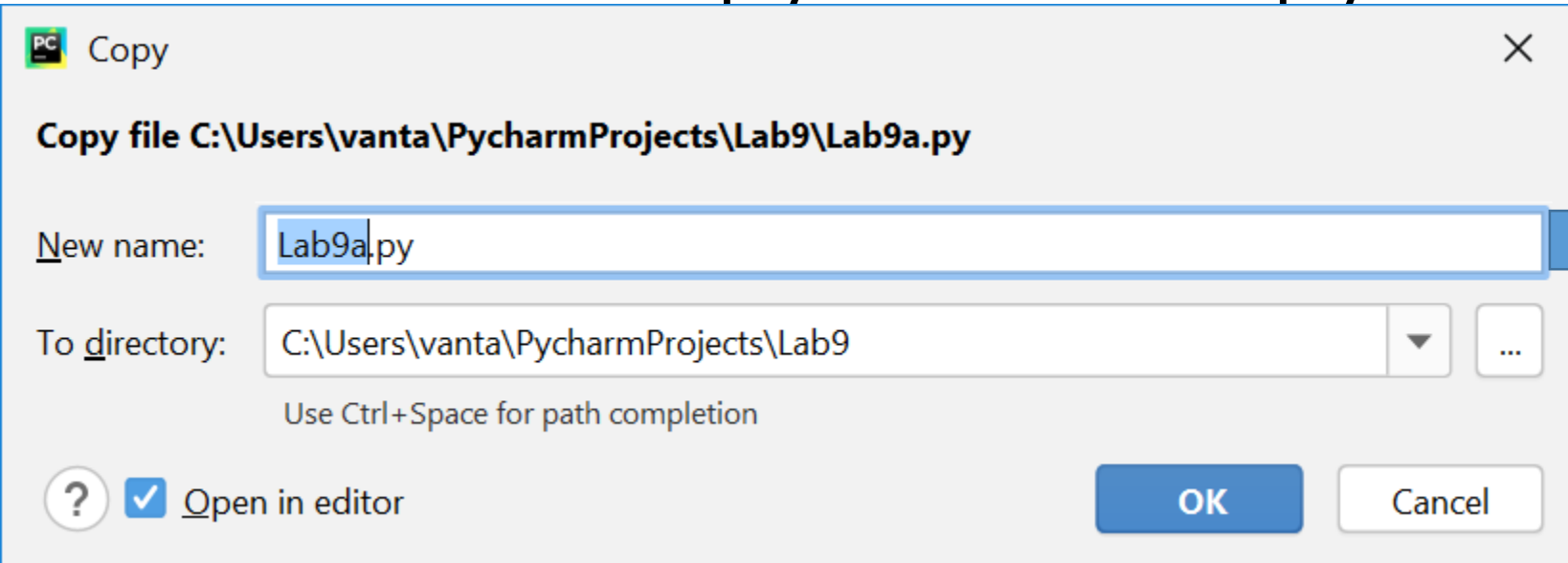
Update ./Lab9/venv/tcl/tcl8.5/init.tcl:

- Fix: package require -exact Tcl 8.5.15

Update ./Lab9/venv/tcl/tk8.5/tk.tcl

- Fix: package require -exact Tk 8.5.15
- You may fix tk.tcl and tcl8.5/init.tcl files in the Python folder

Save Lab9a.py as Lab9b.py



Lab9b.py

```
# importing matplotlib module
from matplotlib import pyplot as plt
def Lab9a():
    # x-axis values
    x = [5, 2, 9, 4, 7]
    # Y-axis values
    y = [10, 5, 8, 4, 2]
    # Create a new figure
    plt.figure(1)
    # Function to plot the bar
    plt.bar(x,y)
    plt.title("Bar graph demonstration")
    plt.show()
```

Lab9b.py

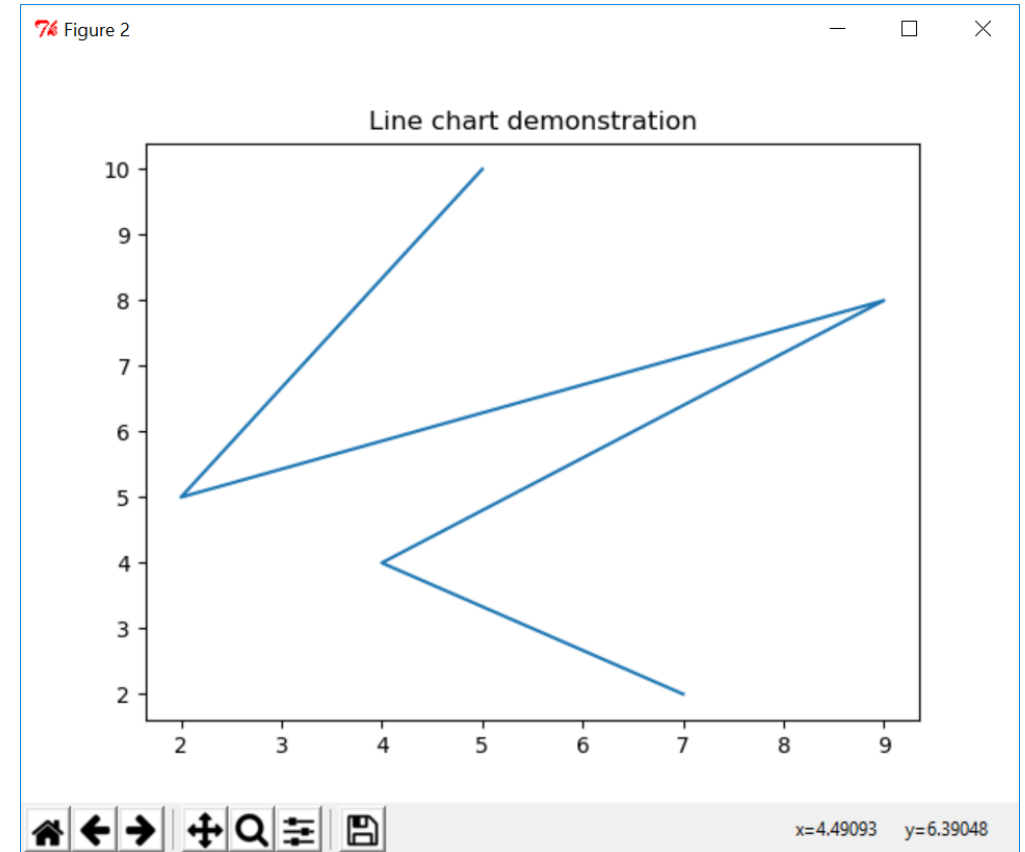
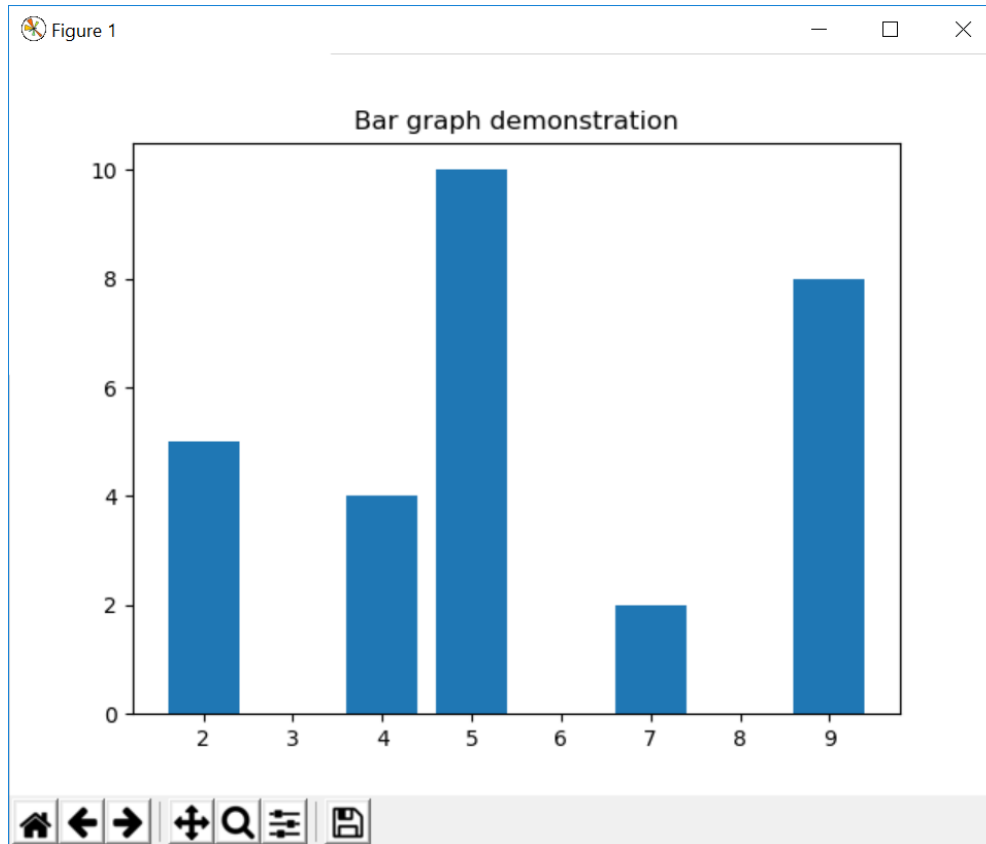
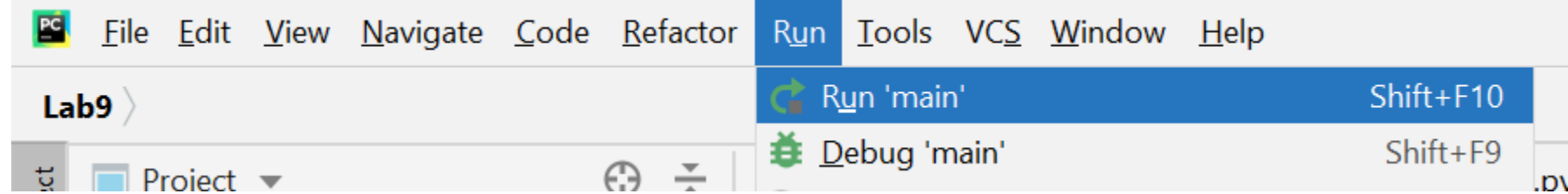
```
# importing matplotlib module
from matplotlib import pyplot as plt
def Lab9b():
    # x-axis values
    x = [5, 2, 9, 4, 7]
    # Y-axis values
    y = [10, 5, 8, 4, 2]
    # Create a new figure
    plt.figure(1)
    # Function to plot the bar
    plt.bar(x,y)
    plt.title("Bar graph demonstration")
    plt.show()
```


Lab9b.py

```
# importing matplotlib module
from matplotlib import pyplot as plt
def Lab9b():
    # x-axis values
    x = [5, 2, 9, 4, 7]
    # Y-axis values
    y = [10, 5, 8, 4, 2]
    # Create a new figure
    plt.figure(1)
    # Function to plot the bar
    plt.bar(x,y)
    plt.title("Bar graph demonstration")
    plt.figure(2)
    # Function to plot the line chart
    plt.plot(x,y)
    # function to show the plot
    plt.title("Line chart demonstration")
    plt.show()
```

main.py

```
from Lab9b import Lab9b  
Lab9b()
```



How can we plot 2 charts in the same figure?

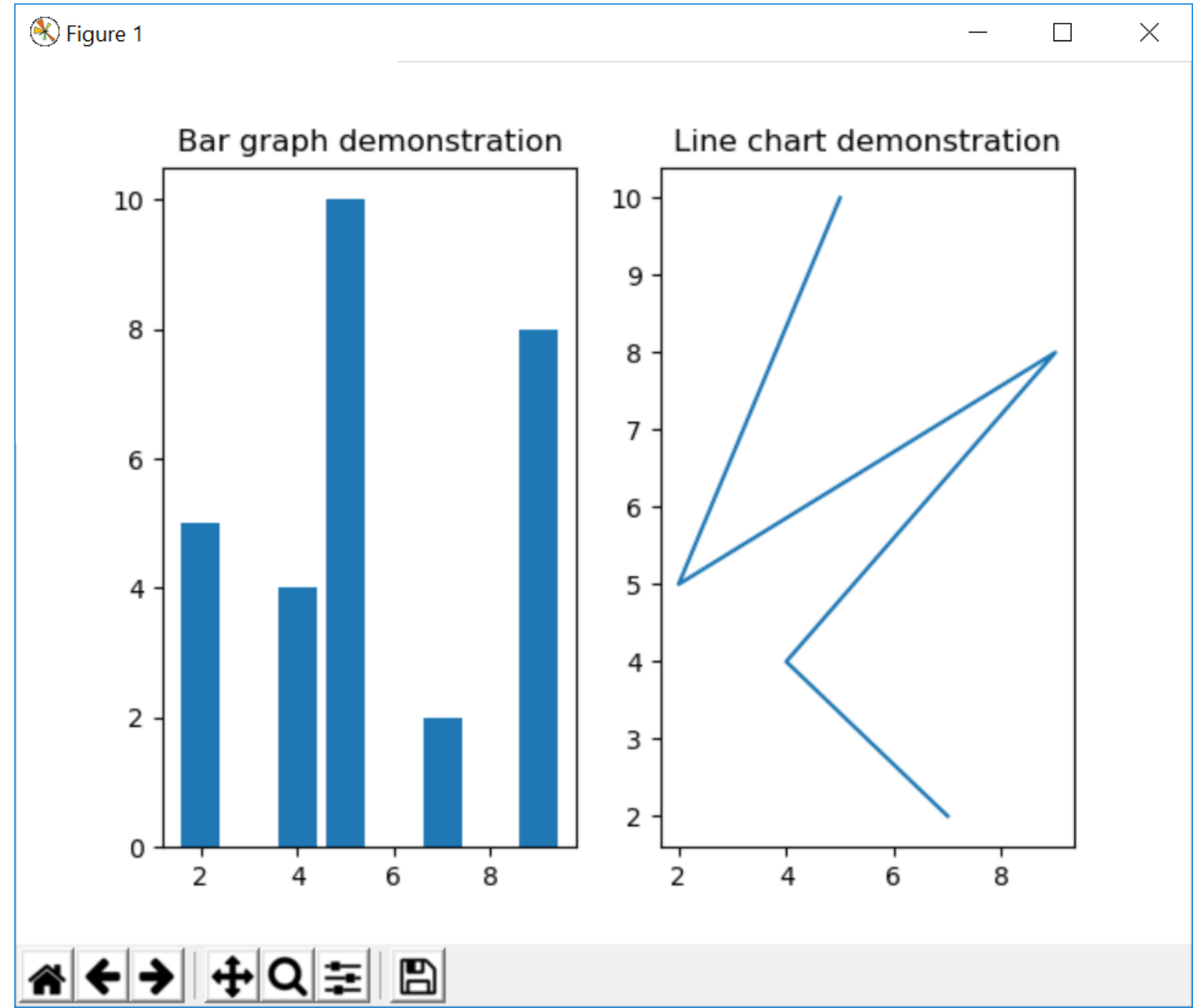
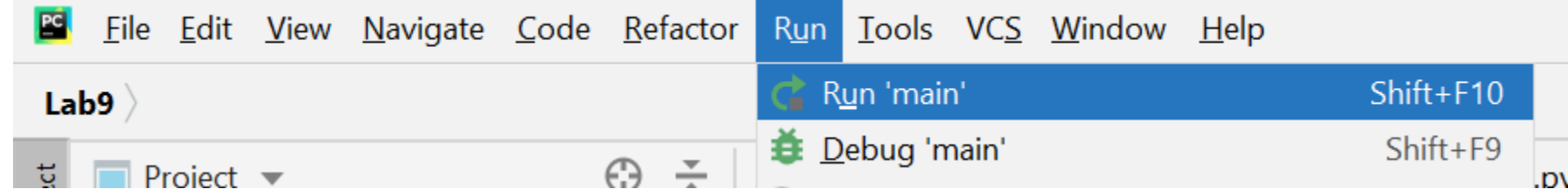
- We can use subplot function

Lab9b.py

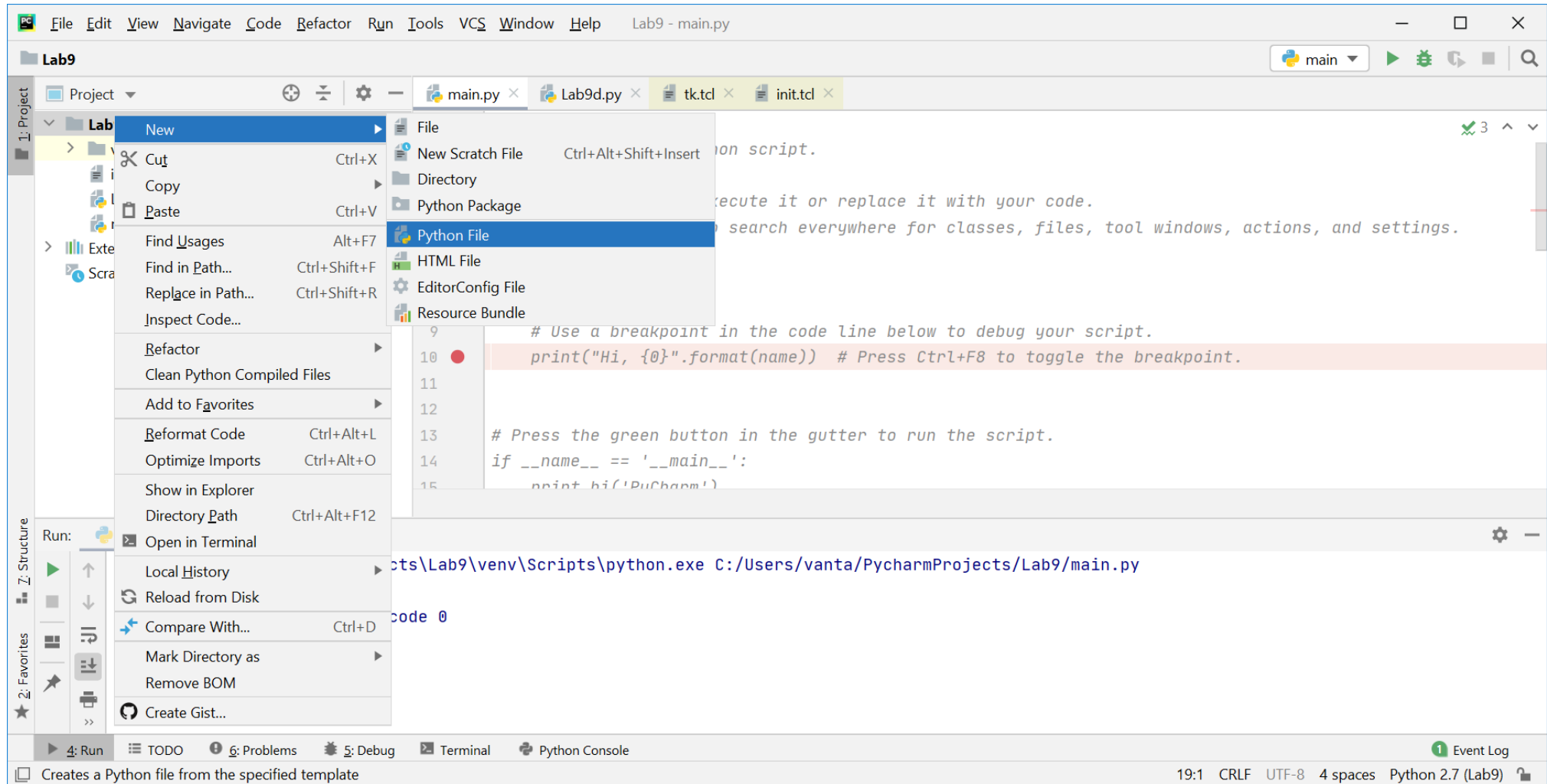
```
# importing matplotlib module
from matplotlib import pyplot as plt
def Lab9b():
    # x-axis values
    x = [5, 2, 9, 4, 7]
    # Y-axis values
    y = [10, 5, 8, 4, 2]
    # Create a new figure
    plt.subplot(1, 2, 1)
    # Function to plot the bar
    plt.bar(x,y)
    plt.title("Bar graph demonstration")
    plt.subplot(1, 2, 2)
    # Function to plot the line chart
    plt.plot(x,y)
    # function to show the plot
    plt.title("Line chart demonstration")
    plt.show()
```

main.py

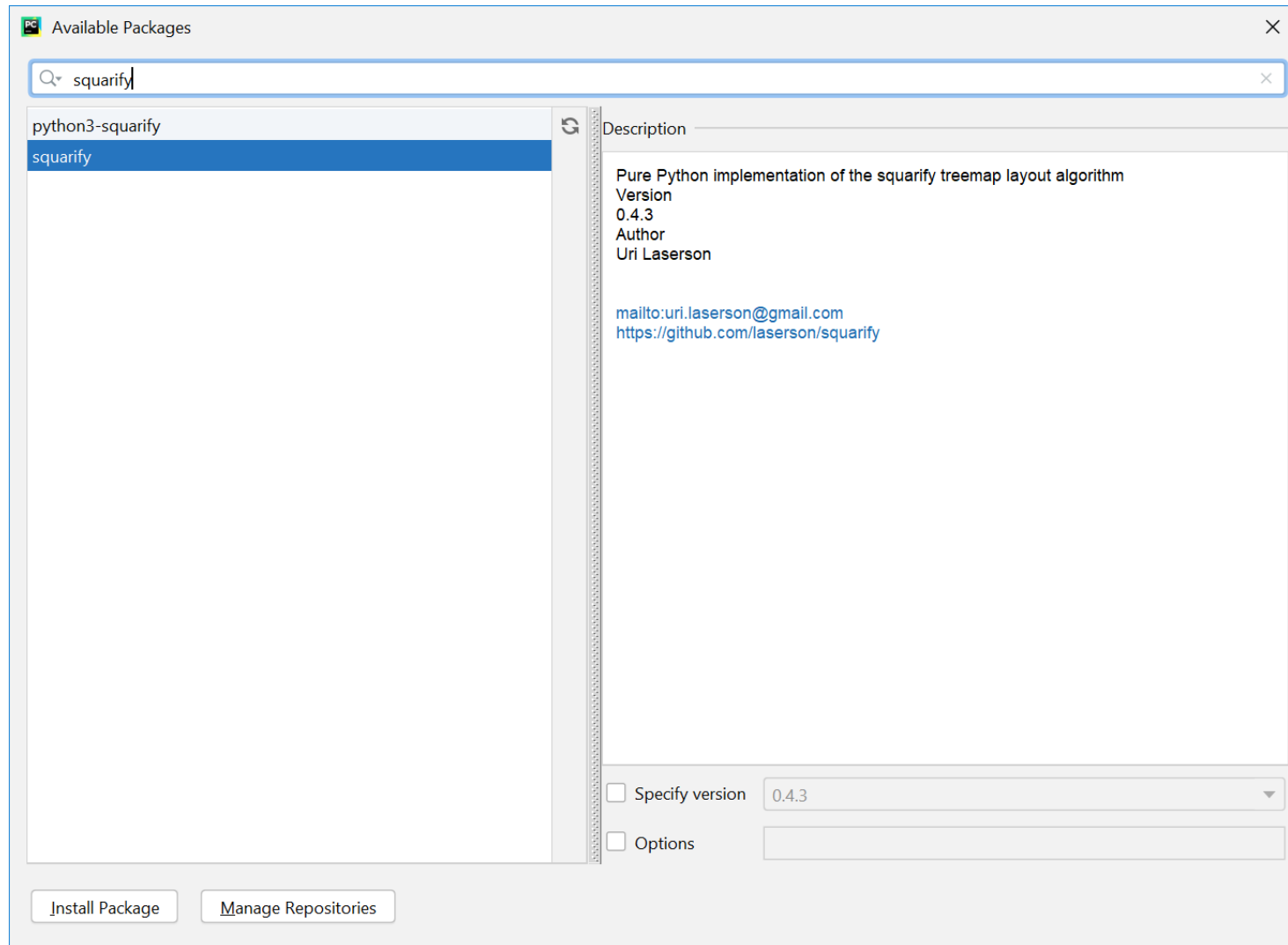
```
from Lab9b import Lab9b  
Lab9b()
```



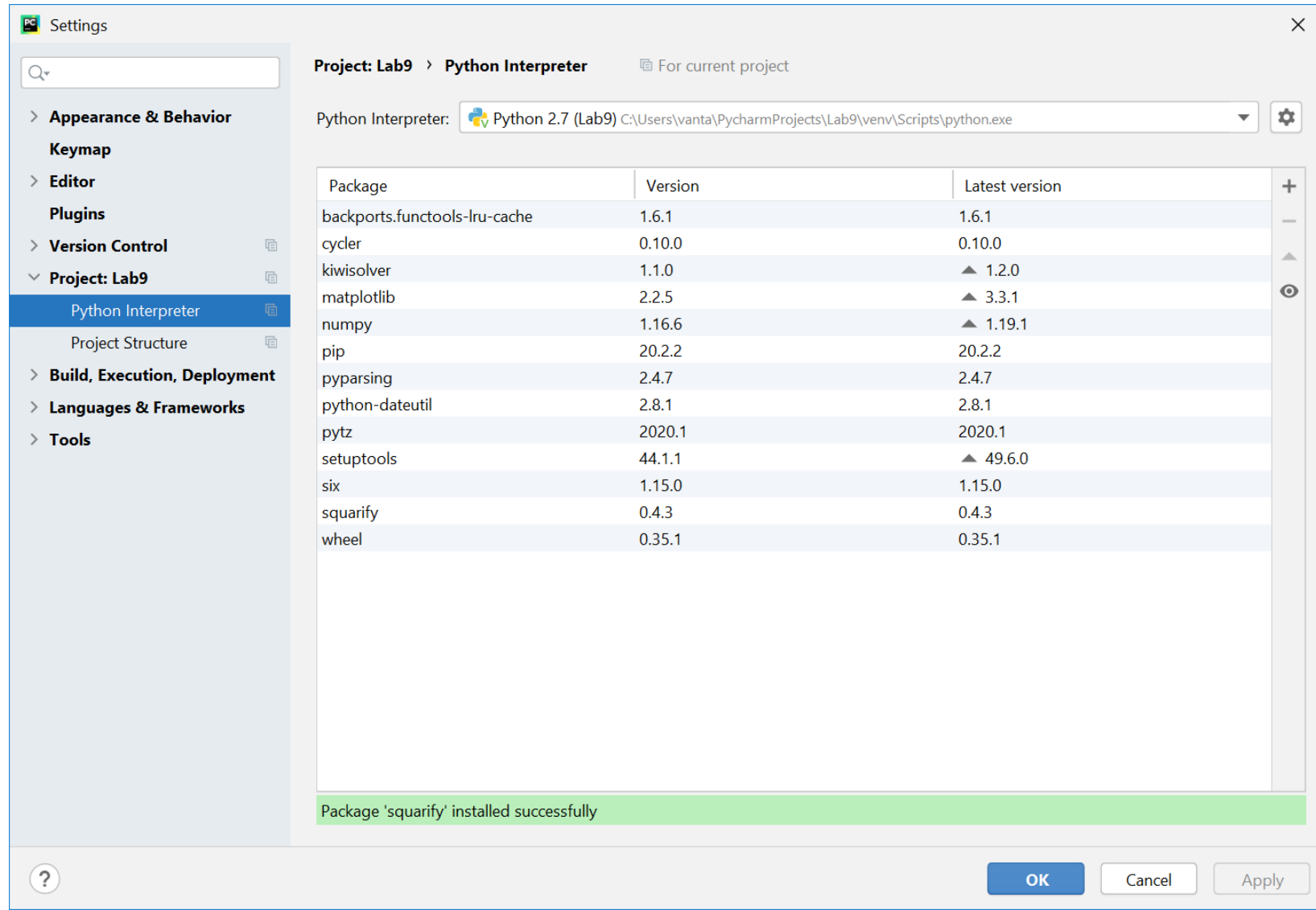
Create Python file: Lab9c.py



Install squarify



Installation finished



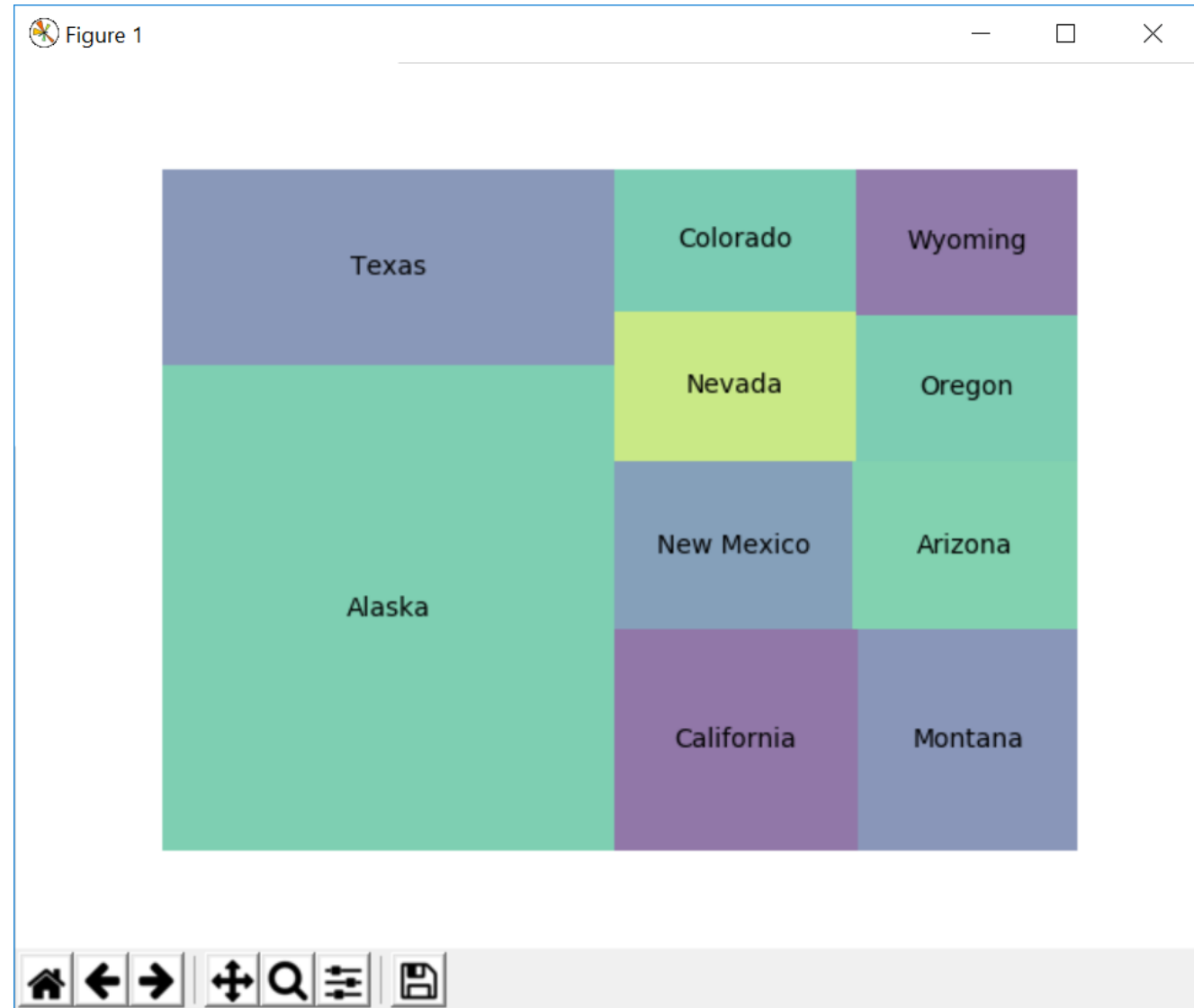
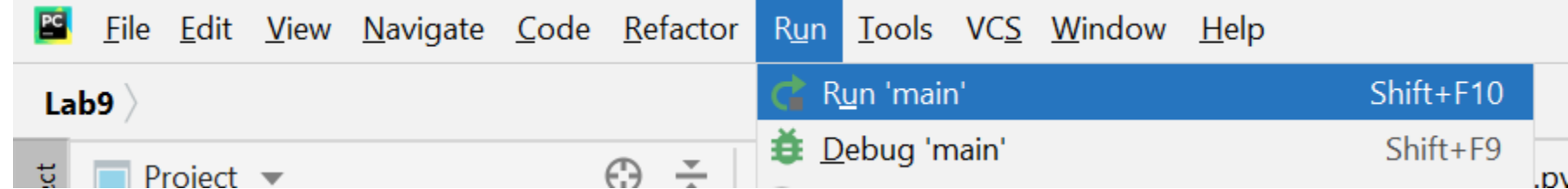
Lab9c.py

```
import matplotlib.pyplot as plt
import squarify
```

```
def Lab9c():
    sizes=[663267.26, 268580.82, 163695.57, 147042.40, 121589.48,
113998.30, 110560.71, 104093.57, 98380.64, 97813.56]
    label=["Alaska","Texas","California","Montana","New
Mexico","Arizona","Nevada","Colorado","Oregon","Wyoming"]
    squarify.plot(sizes=sizes, label=label, alpha=0.6 )
    plt.axis('off')
    plt.show()
```

main.py

```
from Lab9c import Lab9c  
Lab9c()
```



Lab9c.py

```
import matplotlib.pyplot as plt
import squarify
```

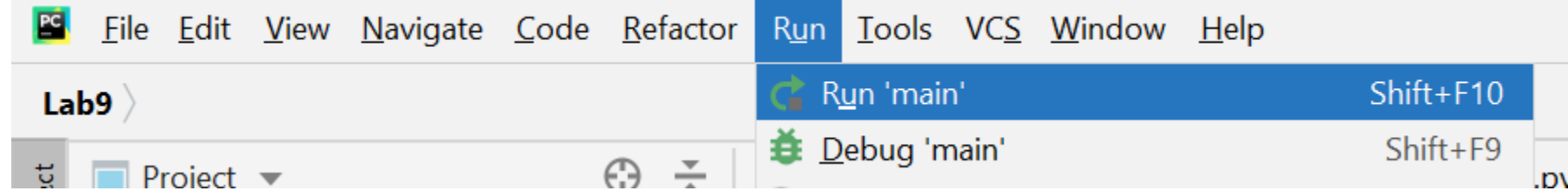
```
def Lab9c():
    sizes=[663267.26, 268580.82, 163695.57, 147042.40, 121589.48,
113998.30, 110560.71, 104093.57, 98380.64, 97813.56]
    label=["Alaska","Texas","California","Montana","New
Mexico","Arizona","Nevada","Colorado","Oregon","Wyoming"]
    color=['red','blue','green','grey','yellow', 'orange', 'brown', 'cyan', 'purple',
'olive']
    squarify.plot(sizes=sizes, label=label, color=color, alpha=0.6 )
    plt.axis('off')
    plt.show()
```

Supported Colors

	black		bisque		forestgreen		slategrey
	dimgray		darkorange		limegreen		lightsteelblue
	dimgrey		burlywood		darkgreen		cornflowerblue
	gray		antiquewhite		green		royalblue
	grey		tan		lime		ghostwhite
	darkgray		navajowhite		seagreen		lavender
	darkgrey		blanchedalmond		mediumseagreen		midnightblue
	silver		papayawhip		springgreen		navy
	lightgray		moccasin		mintcream		darkblue
	lightgrey		orange		mediumspringgreen		mediumblue
	gainsboro		wheat		mediumaquamarine		blue
	whitesmoke		oldlace		aquamarine		slateblue
	white		floralwhite		turquoise		darkslateblue
	snow		darkgoldenrod		lightseagreen		mediumslateblue
	rosybrown		goldenrod		mediumturquoise		mediumpurple
	lightcoral		cornsilk		azure		rebeccapurple
	indianred		gold		lightcyan		blueviolet
	brown		lemonchiffon		paleturquoise		indigo
	firebrick		khaki		darkslategray		darkorchid
	maroon		palegoldenrod		darkslategrey		darkviolet
	darkred		darkkhaki		teal		mediumorchid
	red		ivory		darkcyan		thistle
	mistyrose		beige		aqua		plum
	salmon		lightyellow		cyan		violet
	tomato		lightgoldenrodyellow		darkturquoise		purple
	darksalmon		olive		cadetblue		darkmagenta
	coral		yellow		powderblue		fuchsia
	orangered		olivedrab		lightblue		magenta
	lightsalmon		yellowgreen		deepskyblue		orchid
	sienna		darkolivegreen		skyblue		mediumvioletred
	seashell		greenyellow		lightskyblue		deeppink
	chocolate		chartreuse		steelblue		hotpink
	saddlebrown		lawngreen		aliceblue		lavenderblush
	sandybrown		honeydew		dodgerblue		palevioletred
	peachpuff		darkseagreen		lightslategray		crimson
	peru		palegreen		lightslategrey		pink
	linen		lightgreen		slategray		lightpink

main.py

```
from Lab9c import Lab9c  
Lab9c()
```



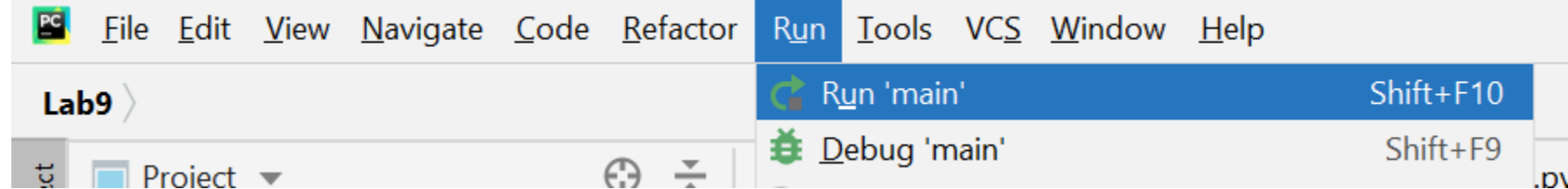
Lab9c.py

```
import matplotlib.pyplot as plt
import squarify
```

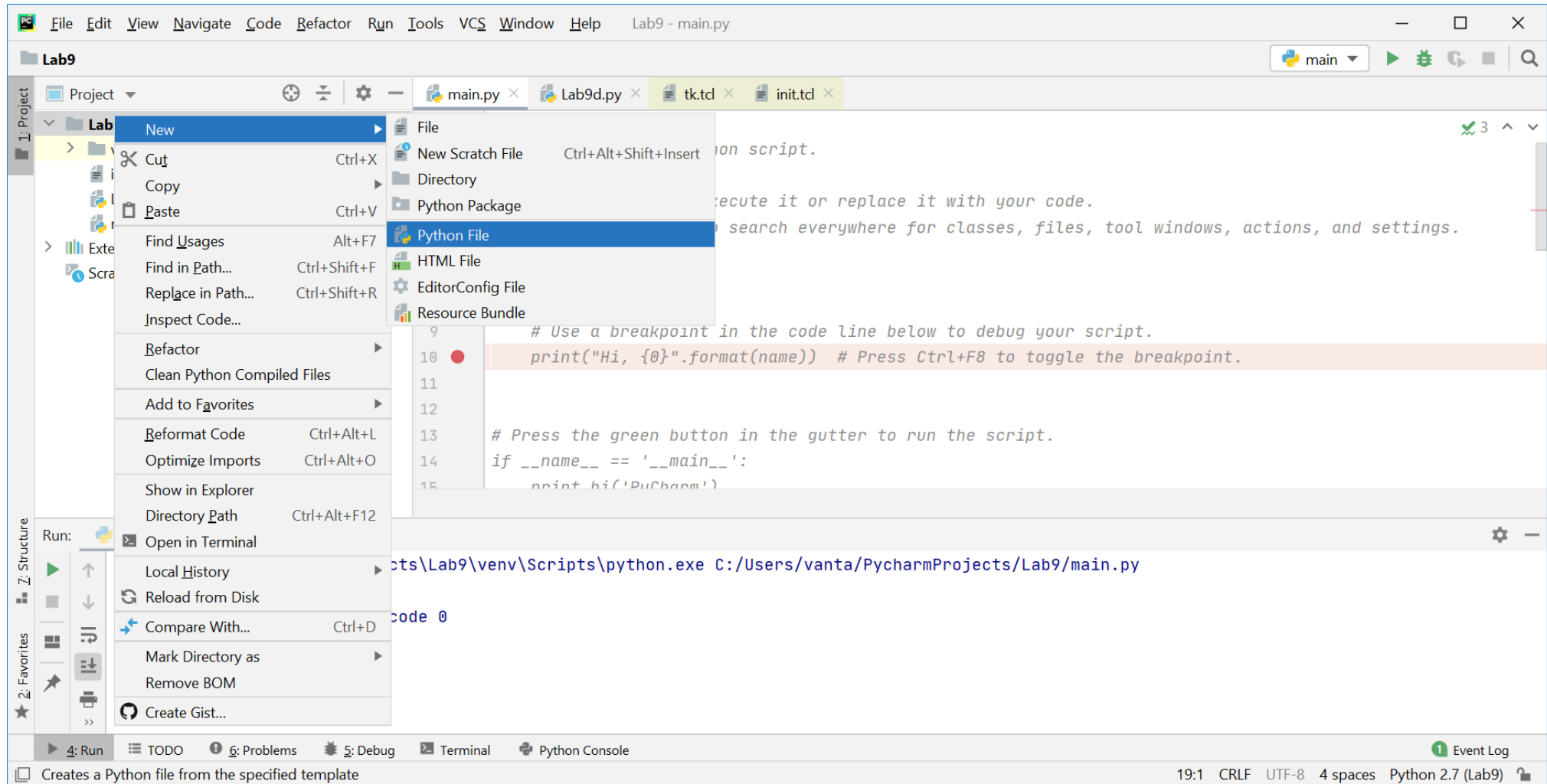
```
def Lab9c():
    sizes=[663267.26, 268580.82, 163695.57, 147042.40, 121589.48,
113998.30, 110560.71, 104093.57, 98380.64, 97813.56]
    label=["Alaska", "Texas", "California", "Montana", "New
Mexico", "Arizona", "Nevada", "Colorado", "Oregon", "Wyoming"]
    color=['red', 'blue', 'green', 'grey', 'yellow', 'orange', 'brown', 'cyan', 'purple',
'olive']
    squarify.plot(sizes=sizes, label=label, color=color, alpha=0.6 )
    plt.axis('off')
    plt.title("Treemap of 10 largest US States")
    plt.show()
```

main.py

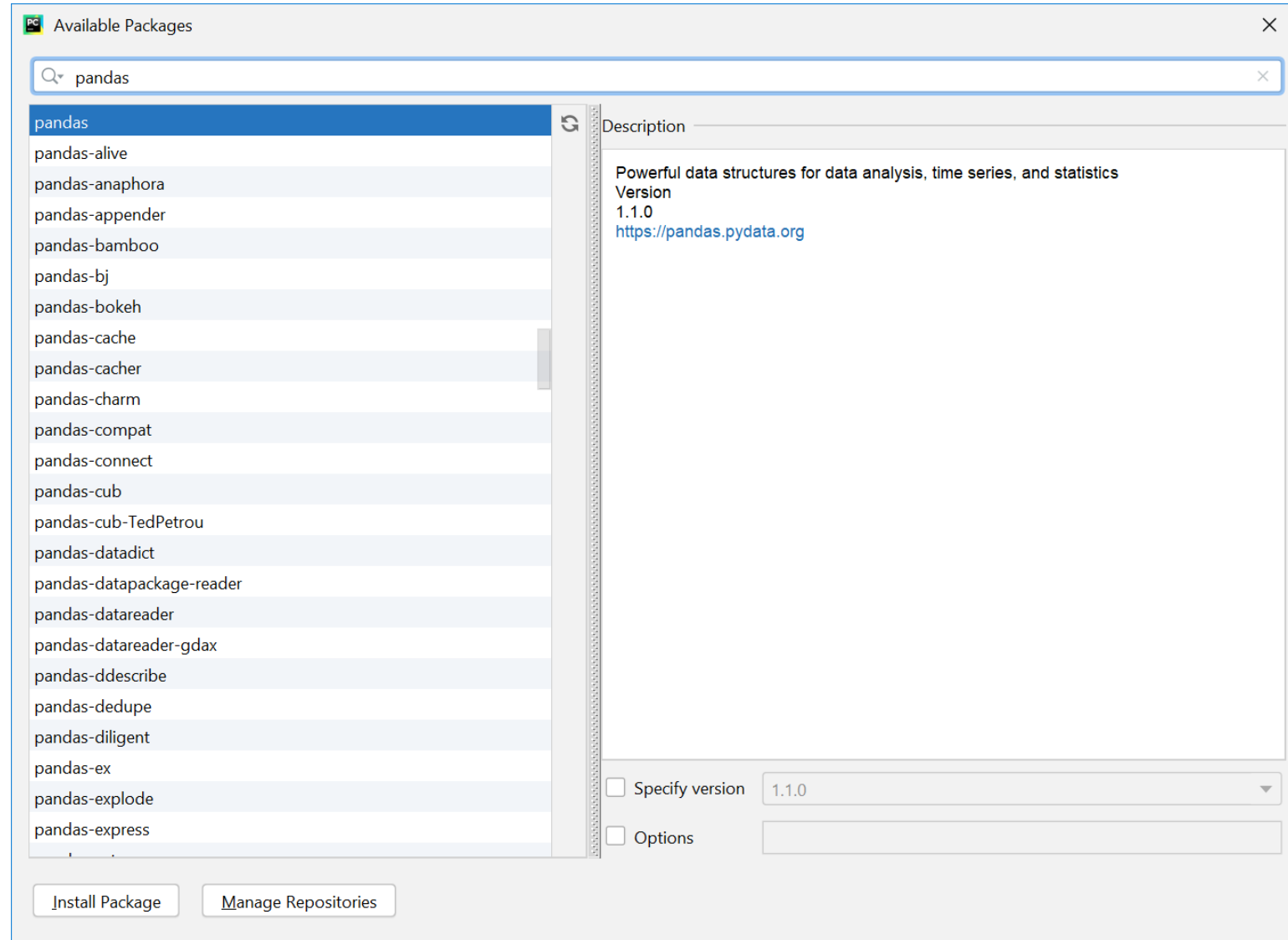
```
from Lab9c import Lab9c  
Lab9c()
```



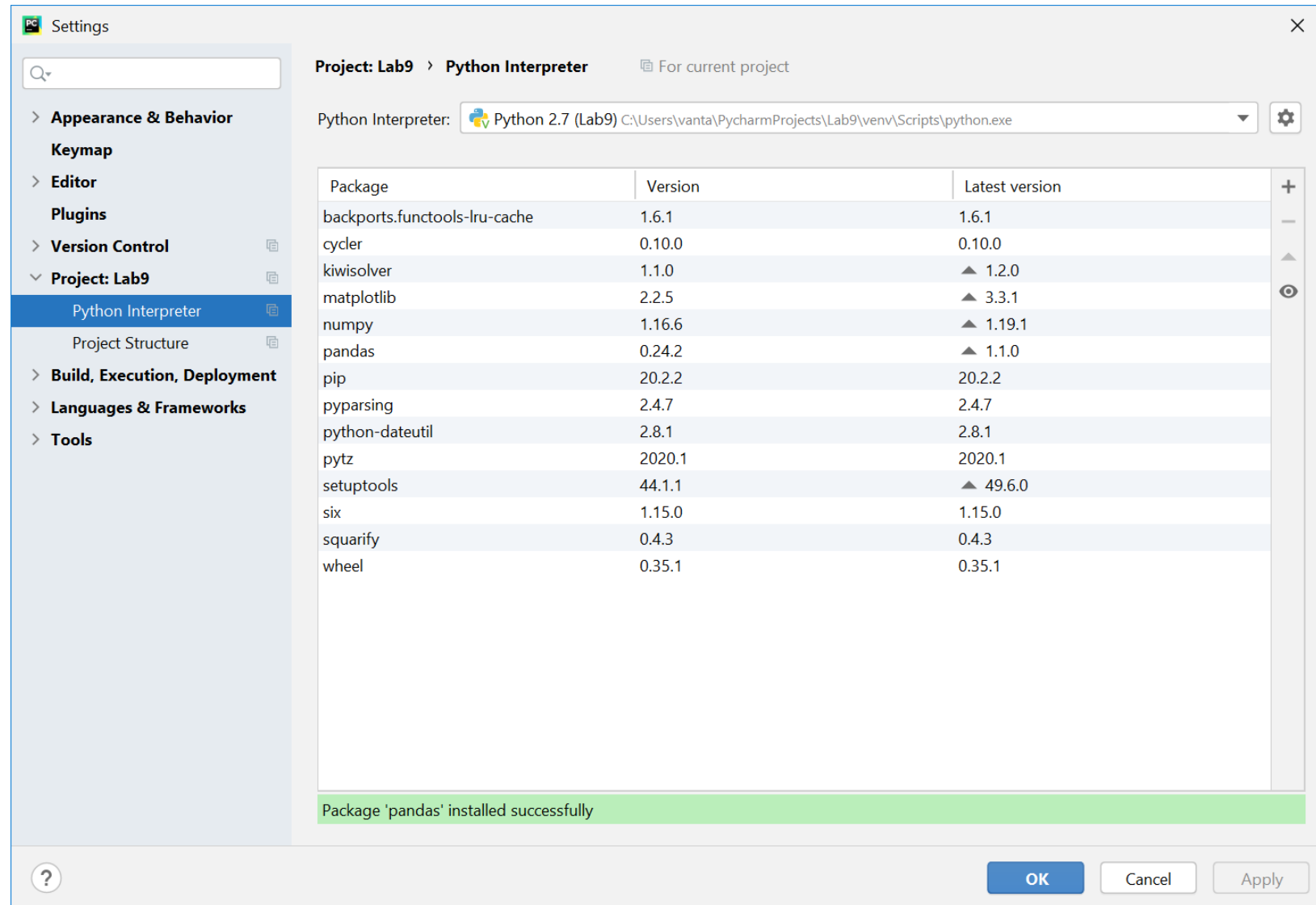
Create Python file: Lab9d.py



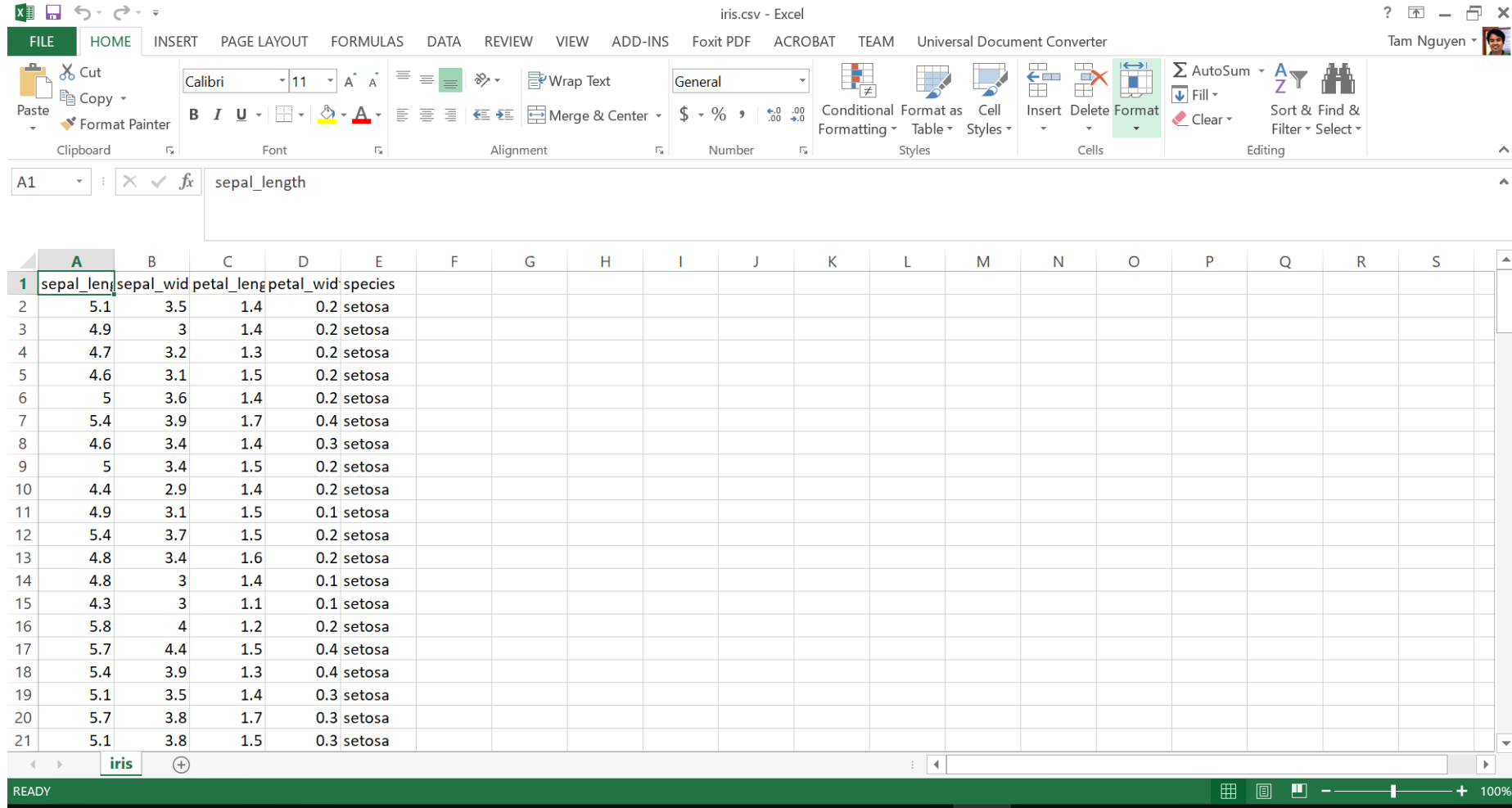
Install pandas



Installation finished



Download iris.csv from isidore



iris.csv - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW ADD-INS Foxit PDF ACROBAT TEAM Universal Document Converter

Clipboard: Cut, Copy, Paste, Format Painter

Font: Calibri, 11, Bold, Italic, Underline, Text Color, Background Color

Alignment: Wrap Text, Merge & Center

Number: General, Currency, Percentage, Decimals, Thousands Separator

Styles: Conditional Formatting, Format as Table, Cell Styles

Cells: Insert, Delete, Format

Editing: AutoSum, Fill, Clear, Sort & Find & Filter, Select

Formula Bar: A1, sepal_length

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	sepal_length	sepal_width	petal_length	petal_width	species														
2	5.1	3.5	1.4	0.2	setosa														
3	4.9	3	1.4	0.2	setosa														
4	4.7	3.2	1.3	0.2	setosa														
5	4.6	3.1	1.5	0.2	setosa														
6	5	3.6	1.4	0.2	setosa														
7	5.4	3.9	1.7	0.4	setosa														
8	4.6	3.4	1.4	0.3	setosa														
9	5	3.4	1.5	0.2	setosa														
10	4.4	2.9	1.4	0.2	setosa														
11	4.9	3.1	1.5	0.1	setosa														
12	5.4	3.7	1.5	0.2	setosa														
13	4.8	3.4	1.6	0.2	setosa														
14	4.8	3	1.4	0.1	setosa														
15	4.3	3	1.1	0.1	setosa														
16	5.8	4	1.2	0.2	setosa														
17	5.7	4.4	1.5	0.4	setosa														
18	5.4	3.9	1.3	0.4	setosa														
19	5.1	3.5	1.4	0.3	setosa														
20	5.7	3.8	1.7	0.3	setosa														
21	5.1	3.8	1.5	0.3	setosa														

READY 100%

Lab9d.py

```
import pandas
import matplotlib.pyplot as plt
from pandas.plotting import parallel_coordinates

def Lab9d():
    data = pandas.read_csv('iris.csv', sep=',')
```

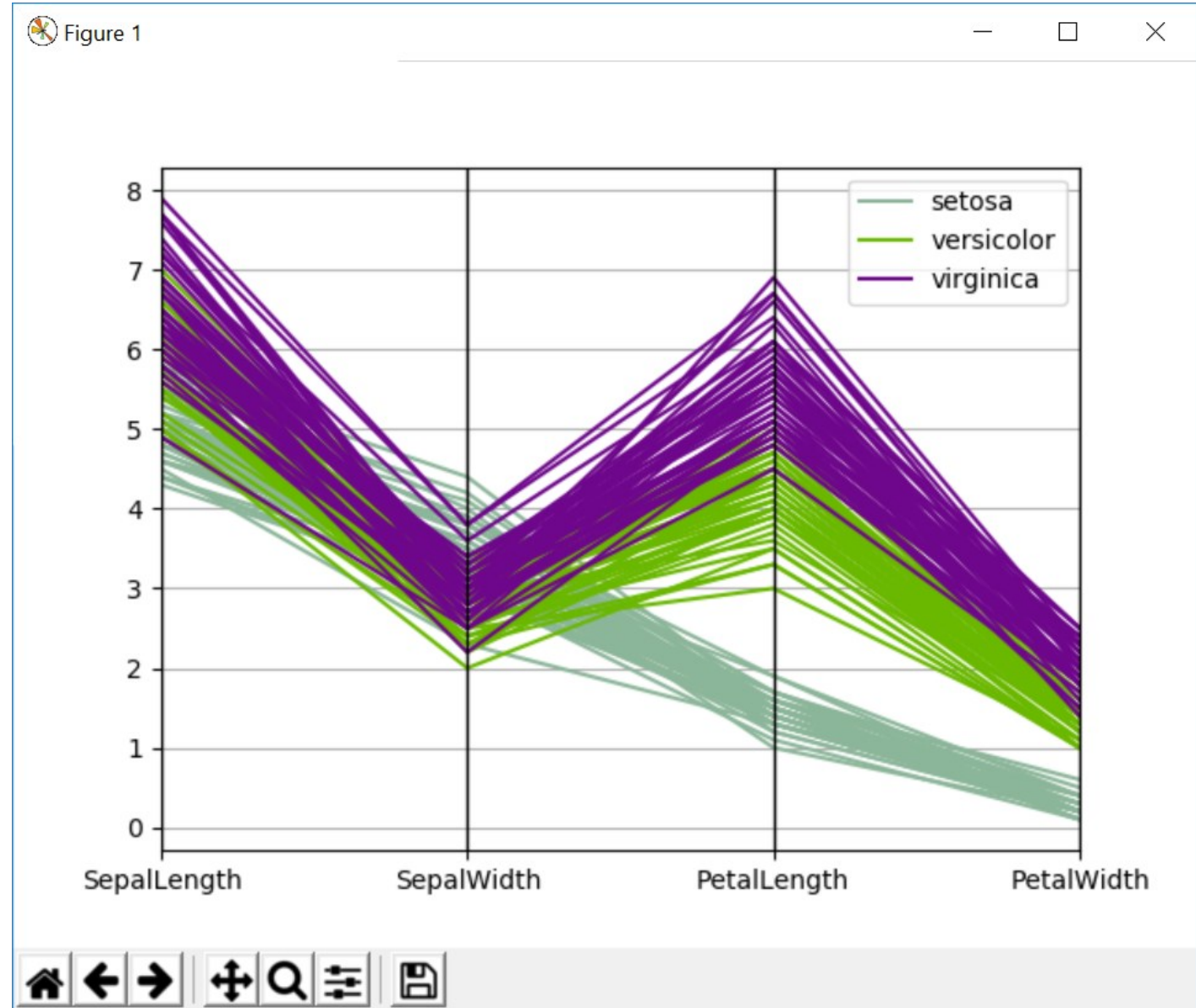
Lab9d.py

```
import pandas
import matplotlib.pyplot as plt
from pandas.plotting import parallel_coordinates
```

```
def Lab9d():
    data = pandas.read_csv('iris.csv', sep=',')
    parallel_coordinates(data, 'Name')
    plt.show()
```

main.py

```
from Lab9d import Lab9d  
Lab9d()
```



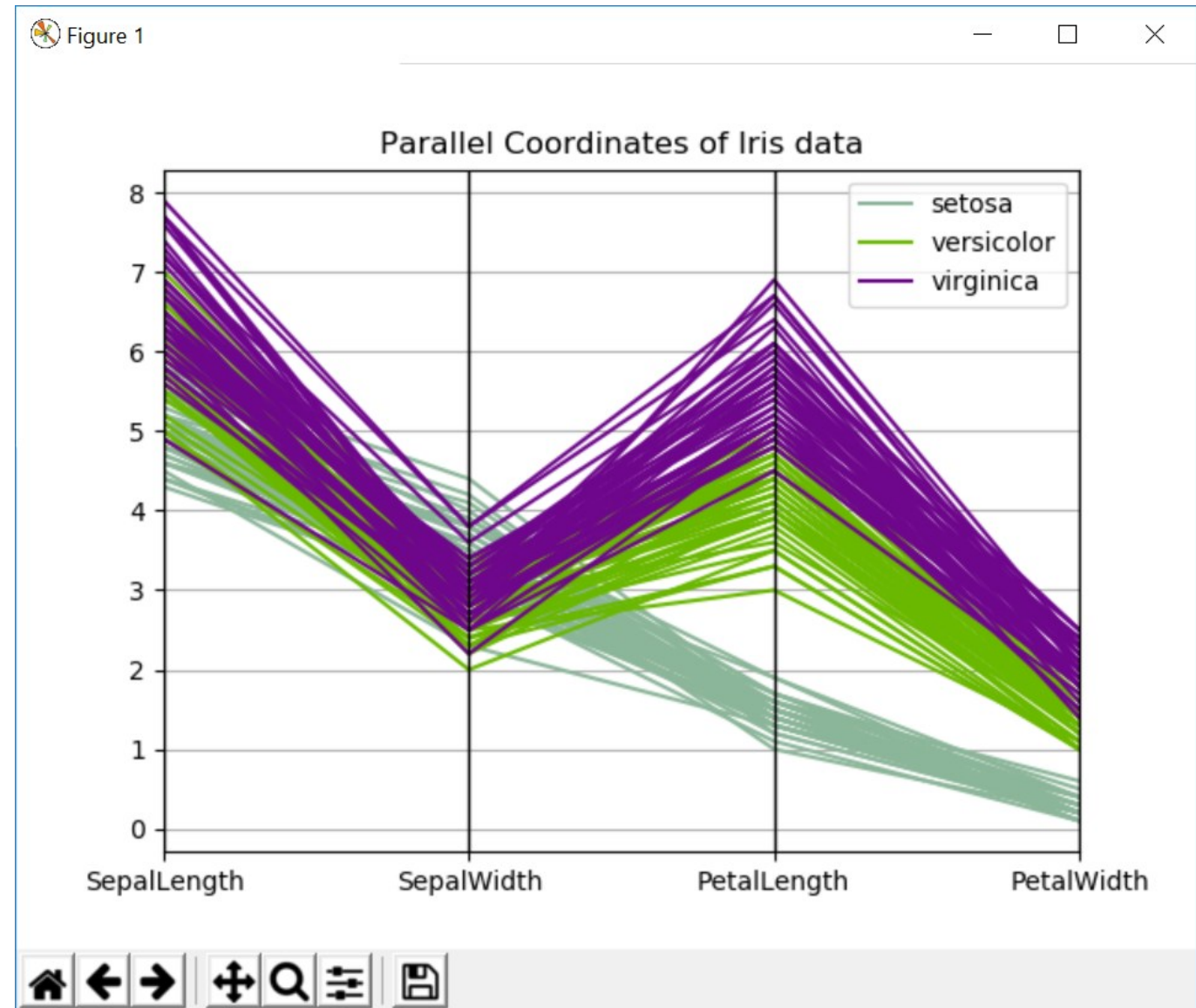
Lab9d.py

```
import pandas
import matplotlib.pyplot as plt
from pandas.plotting import parallel_coordinates
```

```
def Lab9d():
    data = pandas.read_csv('iris.csv', sep=',')
    parallel_coordinates(data, 'Name')
    plt.title("Parallel Coordinates of Iris data")
    plt.show()
```

main.py

```
from Lab9d import Lab9d  
Lab9d()
```



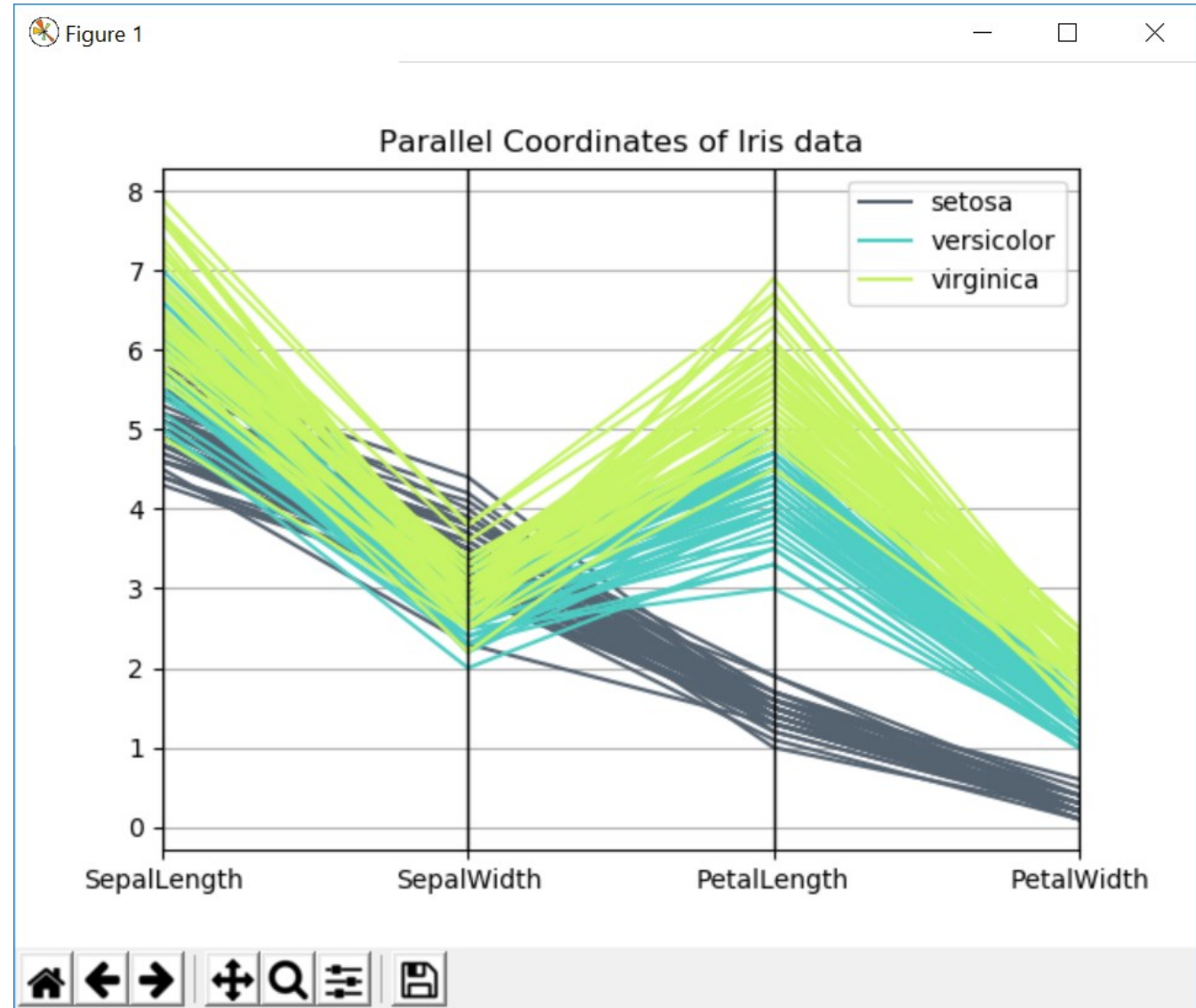
Lab9d.py

```
import pandas
import matplotlib.pyplot as plt
from pandas.plotting import parallel_coordinates
```

```
def Lab9d():
    data = pandas.read_csv('iris.csv', sep=',')
    parallel_coordinates(data, 'Name', color=('#556270',
'#4ECDC4', '#C7F464'))
    plt.title("Parallel Coordinates of Iris data")
    plt.show()
```

main.py

```
from Lab9d import Lab9d  
Lab9d()
```



Q&A