**Final Project and Exam**

The primary focus of this module is to practice the skills gained earlier in the course and then demonstrate those skills in your final assignment. For the final assignment you will analyze historical automobile sales data covering periods of recession and non-recession. You will bring your analysis to life using visualization techniques and then display the plots and graphs on dashboards. Finally, you will submit your assignment for peer review and you will review an assignment from one of your peers. To wrap up the course you will take a final exam in the form of a timed quiz.

**Learning Objectives**

* Practice creating a dashboard
* Practice visualization skills
* Review and grade an assignment submitted by peers
* Create a dashboard and add interactivity
* Create various visualizations using a number of plot libraries

Hide Learning Objectives

**Practice Project**

# **Final Project**

(Optional)Reading: Common Issues in Dash Application

**Key considerations for developing a Dash application**

1. Please ensure you have added the correct column values as indicated in the hint for each code snippet. Example for **X and Y** in the charts
2. Please ensure that you have used the correct **indentation** in the code.
3. Fill in your code using the **placeholders(….)** provided in the code skeleton provided in the lab. Separate each component by commas.

**Example:** This creates a **Div element in HTML**, a container for other components. Inside this div, various child elements are placed, which need to be separated by commas.

10

**Common errors:**

**1. Syntax error: invalid syntax def update\_input\_container(selected\_statistics):**

**Solution:** Check if any parentheses or square brackets are unclosed before the **def** function.

The HTML components are organized in a hierarchy using brackets. The use of square brackets **[] within html.Div()** or any other Dash component is essential for defining children elements in a list. This list-based structure allows you to include multiple child components inside a single-parent component.

**Outer HTML**

1

2

3

4

html.Div([ ...

            ...

        ])

This html.Div is the container for the entire layout of the dashboard. It encapsulates and structures all the other HTML elements as a cohesive unit.

They use square brackets to group multiple child components within a single-parent component. This enables you to structure complex layouts with nested elements.

Dash components need to maintain a clear hierarchy to render correctly. The square brackets help maintain this hierarchy by clearly indicating which components are children of a particular parent component.

**Inner HTML**

1

2

3

4

5

6

7

8

9

html.Div([

        html.Label("Select Statistics:"),

        dcc.Dropdown(

                    id='...........',

                    options=...................,

                    value='.................',

                    placeholder='.................'

                     )

     ]),

**2. When encountering a Callback error updating output. container.children**

**Solution:** Ensure **Input variables** used within the **callbacks** are defined and correctly scoped in the correct order.

1

2

3

4

5

@app.callback(

Output(component\_id='output-container', component\_property='children'),

[Input(component\_id='dropdown-statistics', component\_property='value'),

Input(component\_id='select-year', component\_property='value')])

**3. Graph not displaying.**

**Solution:** Ensure you have used the **elif** statement with the correct variable.

1

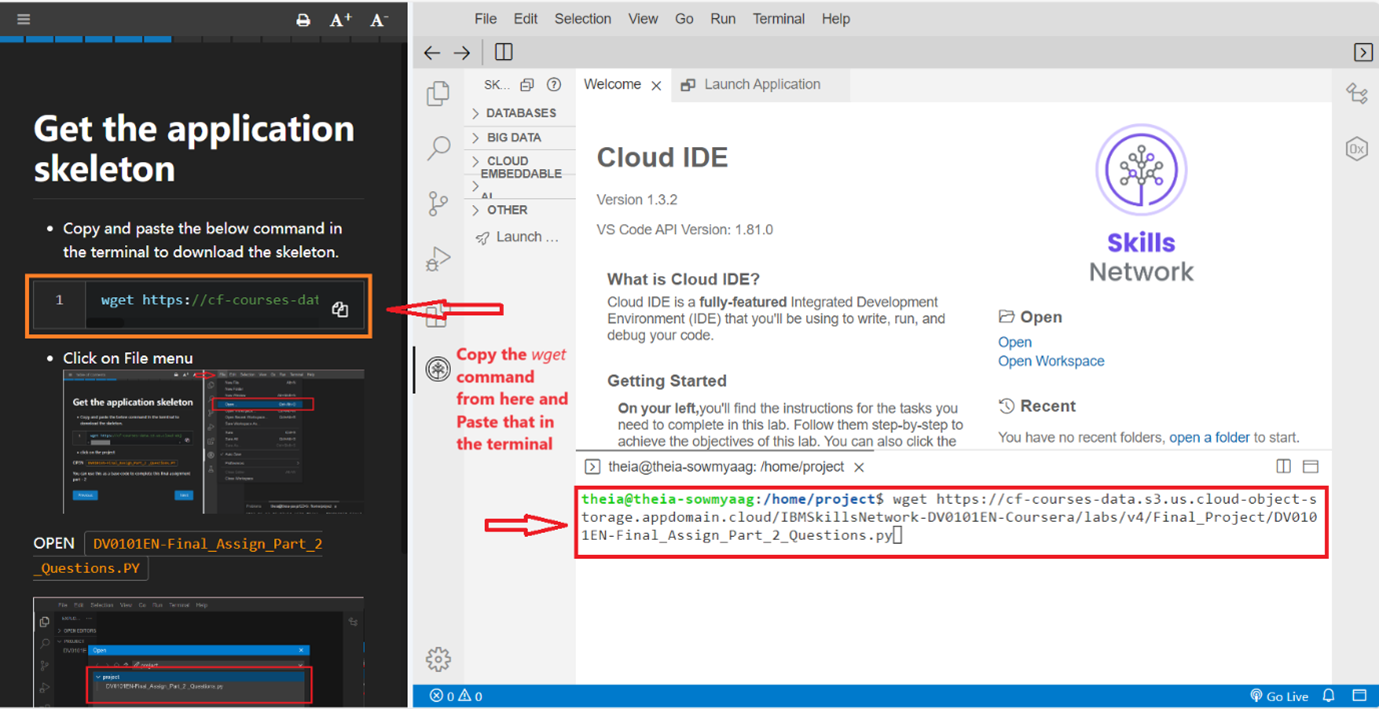
2

elif (input\_year and selected\_statistics=='...............') :

        yearly\_data = data[data['Year'] == ......]

**4. The file could not be located under the project.**

**Solution:** Copy the **wget** command and paste it into the **terminal** to execute. The file will then be in the **Project directory.**



**5. Port 8050 is in use by another program. Either identify and stop that program or start the server with a different port**

**Solution:** Try with different port as shown below:

1

2

3

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(port = 8090)

Mark as completed

Like

Dislike

# Report an issue Course Summary

In this course, you learned how to use Data Visualization with Python to create charts and graphs for business or educational purposes. Graphs and charts are essential for communicating statistical information directly and efficiently. In this course, you also learned the history of Data Visualization with Python, why it was created, and who created it. Understanding the creation process of Data Visualization applications and programs is essential for understanding its uses, abilities, and importance. You can apply these skills to other graphing and statistical visualization applications that utilize Python and code.

**Next Steps**

After completing this course, you could take similar courses related to Data Visualization with Python, “Python for Data Science AI & Development, Python for Data Science, and Databases and SQL for Data Science with Python courses.”

Congratulations and Next Steps

Congratulations on completing the Data Visualization with Python course! You are one step closer to preparing for theIBM Data Science Professional Certificate, IBM Data Analyst Professional Certificate, and Applied Data Science Specialization.

As you have learned, Data Visualization with Python can be essential in many career paths. In this course, you were introduced to Data Visualization, including useful programs, graphs, models, and methodologies to utilize Data Visualization with Python. You also learned how Data Visualization with Python can be a valuable tool for researching, graphing, and coding.

We hope this course is just the beginning of your journey to learning more about Data Visualization with Python. If you want to continue learning, please check out our [Python for Data Science AI & Development](https://www.coursera.org/learn/python-for-applied-data-science-ai?specialization=ibm-data-science), [Python for Data Science](https://www.coursera.org/learn/python-project-for-data-science?specialization=ibm-data-science), and [Databases and SQL for Data Science with Python](https://www.coursera.org/learn/sql-data-science?specialization=ibm-data-science) courses.

# We also encourage you to leave your feedback and rate this course so we can continue to improveThanks from the Course Team

The entire course team thanks you for taking this course. We hope you enjoyed it and wish you the best in applying your new knowledge and skills.

We encourage you to rate the course and provide a review. Your feedback is much appreciated.

Best regards,

Course Team

- - -

This course has been brought to you through the involvement of the following team of contributors:

**Primary Instructors**

* Saishruthi Swaminathan

**Other Contributors and Staff**

* Project Lead: Rav Ahuja
* Instructional Designer: Madhusudan M Rao, Andrew Pfieffer
* Subject Matter Expert: Dr. Pooja
* Lab Authors: Dr. Pooja

**Production Team**

* Publishing: Rachael Jones, Grace Barker
* QA: Dawn Teel-Friedman, Mercedes Schneider, Mary Stenberg, Steve Hord, Beth Larsen, Bethany Hudnutt, Sydney Marsing
* Project Manager: Alison Woolford
* Narration: Gale Gruz
* Video Production: Prayas Gupta and team

**Teaching Assistants and Forum Moderators**

* Teaching Assistants: Malika Singla, Lakshmi Holla, Akansha Yadav, Niveditha Pandith T S, Sathya Priya

our content. Thank you!