Note: To use this file, click File > Download as and choose your file type.

Rhyme Project Script Template

Read more about [producing engaging videos](https://partner.coursera.help/hc/articles/203525739-Producing-Engaging-Videos) in the [Partner Resource Center](https://partner.coursera.help/hc).

**Core Information:**

* Instructor Name: David Dalsveen
* Project Title: Data Visualization with Python

|  |  |
| --- | --- |
| **Project Description:**  *What hands-on artifact will learners have developed by the end of this project? Why will they want to know how to do this? Give a 200-300 word description of the project. This will appear on the Coursera platform.* | Visualizing data is used by virtually every discipline these days. It is used for analyzing web traffic to determine peak server load, growth and death rate of populations for biological analysis, analyzing weather patterns over time, stock market trends, and so on. Simply put, Data Visualization brings meaning to numbers that help people understand it. Seeing the data change can draw attention to trends and spikes that may otherwise go unnoticed. Python is an open-source (free) programming language has libraries that can be used to read and make useful graphics to present the data.  In this course, you will create an application that reads data from CSV files. You will learn how to visualize the data using various techniques using existing Python libraries. |
| **Target Learner:** *Who will benefit from this project? Consider the job roles the learners currently perform, where they want to go in their career, how this project will benefit their goals, etc.* | This project will be a good starting point for learners who want to gain a good foundation with Python data visualization. It will demonstrate how to use Python to read and visualize data, which is an important part of data science. |
| **Project Prerequisites:** *List any knowledge or skill prerequisites needed to complete the project (e.g. familiarity with matplotlib or Excel spreadsheets)* | Learners should be familiar with some Python, including variables and input/output. Learners should also be familiar with using an IDE such as Visual Studio Code to create and launch projects. |
| **Number of Tasks in Project:** *Each Task is a Video in a Rhyme project. Each Task should have a clear learning objective, where the learner follows along in a hands-on way. Those learning objectives should contribute to the final artifact they are working toward.* | Five |
| **Expected Project length (mins):**  *The whole project, so all Tasks combined, should be <1 hour total. Estimate 150-180 words per minute of video per Task. Suggested length is 4-7 minutes per video.* | 55 |
| **Software needed for Rhyme VM instance:** | Windows, Visual Studio Code, Chrome Browser (for reference) |

**Project Learning Objectives:**

Before you design the Storyboard, list out the project objectives. “By the end of each Task, learners will be able to…”

* LO1: Use pandas to produce a table to visualize the data.
* LO2: Produce a data plot using MatPlotLib Pyplot.
* LO3: Use Seaborn to create a scatterplot graph.
* LO4: Create a heatmap to show correlations and distributions.
* LO5: Create a Jointplot to show distribution.

Task Names for Rhyme and Coursera: Create a Pandas DataFrame Table, MatPlotLib Plot, Scatterplot Graph, Heatmap plot, Jointplot Distribution

Course Objectives: Create DataTable, Plot data with MatPlotLib, Use Seaborn to visualize data.

**Storyboard**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section/**  **Scene** | **Learning Objective** | **Script** | **What is the visual for this Task? This appears in the VM instance** | **Draft ideas to create quiz questions to test the core concepts for this Task** |
| Task 1 | LO1 | <SLIDE> Hello and welcome to our course on Data Visualization in Python.  I’m Dave Dalsveen, your instructor for this course.  In this course, you will create an application that reads data in a couple of different formats using the Pandas library. You will produce a table based on a dataframe to visualize the data.  You will also  You may already know that Python is one of two popular languages in the field of data science today. Virtually every major company including Google, Amazon, and IBM is involved in data science in some way. A part of the field of data science is the ability to visualize data, to make sense of data in human terms.  If you have used Python before, you realize there are a lot of prebuilt libraries to help with application development. The Pandas MatplotLib and Seaborn are libraries essential to data visualization with Python.  A popular data structure to use in Pandas is the DataFrame, due to its usefulness and simplicity. Because it is two dimensional, it is organized into rows and columns just as a spreadsheet is.  <SLIDE>  You will notice that there are two panels on your screen. I’ll be working on the panel on your right. That is where I’ll demonstrate and cover your lesson material. Your work will take place on the panel on the left. There, you can work along with me and get real hands-on practice that makes this training so effective. Remember, you can pause the recording at any time to type in the code, run the code, and make corrections as we go.  Now, let us get started!    In this project, we will be staring with clean data sets represented in a json file and in CSV files. Using the Pandas, Matplotlib, and Seaborn libraries, we will visualize temperature and rainfall data from a particular region and see if the data are correlated in any way. We will also visualize birth data by year from a particular country to see when the most births occurred(year and month).   1. You will see Visual Studio Code open with the PythonDataV loaded. 2. In the code window, you will see the Python file already loaded. The file is called task1.py as you can see. We will be working in this file in this project, and then switch to a different file for each task. This will help to keep the code from getting too cluttered. 3. Go into the code window and type: import pandas as pd The reason we use ‘as pd’ is so that we can access the Panda library classes using the handle ‘pd’ in the program. 4. Next, import matplotlib.pyplot as plt This will allow us to show the plot we will create in this lesson using the dataframe. 5. Next, create the dataframe itself. Type: #Create a DataFrame using the json data   df = pd.read\_json (r'./rain.json')   1. Next output the DataFrame, which, it turns out is a nice representation of the data in table form: Type: #output the dataframe print (df) 2. To get more insight into the data, we can also print some common statistics, all in one command using describe, Type: *print("df statistics: " ,df.describe())* 3. Next, let us use the scatter plot method of the DataFrame plot object. *df.plot.scatter(x='Rainfall', y = 'Temperature')* 4. It is important that the x and y assignments match what is in the DataFrame table column headings. 5. Next, we can add a Label to identify what each dot means. Add Label before the closing parenthesis: label="Rainfall and Temperature" *df.plot.scatter(x='Rainfall', y = 'Temperature', label="Rainfall and Temperature")* 6. Finally, we need to show the plot in a window. We will use the show method of PyPlot to create a window and show the resulting plot done by our dataframe!  Type: *plt.show()*   In this course, we concentrate on giving you a foundation in visualizing the data using MatPlotLib and Seaborn. Throughout the remaining lessons, we will read the data from a file, and see what tools Python gives us to visualize the data. Here is an example of one of our upcoming plots:  <pause here>  Great, now that you have had a taste of how we can visualize data using the DataFrame, let us move on to the next lesson.  In the next lesson, we will go over visualizing the data with MatPlotLib’s pyplot plotting functions. | The Learner will learn how to create a visual table of data based on the dataframe. | How do we create a table from the dataframe using a .csv file? |
| Task 2 | L02 | Welcome back! Now that we know how to use a Dataframe to retrieve data and plot it, let us see how use PyPlot to plot the data.  Remember, you can pause the recording at any time to analyze and type in the code, run the code, and make corrections..  First, we will open the file called task2.py.   1. Open the file called task2.py. 2. Add the same imports as we did in task1.  Namely: import pandas as pd   import matplotlib.pyplot as plt.   1. Next, populate and print the DataFrame so that we have a visual reference! # 1 create the dataframe using the json file..   df = pd.read\_json (r'./rain.json') print((df)   1. Next, let us try to plot the data using polyplot. We will plot the temperature according to month. We will need to extract the appropriate fields for the X-axis (Month), Y-Axis(Temperature). Type: plt.plot( df['Month'], df['Temperature'], label='Temperature') 2. Next, use polyplot’s show method to create the window. Type: *plt.show()* 3. The graph looks great, except it is not large enough to see all of the months. Before the plt.plot code, change the size by typing the following: *plt.figure(figsize=(15,5))* 4. Now the frame is the correct size. You notice the X axis is labeled with months, so it’s meaning is quite obvious. What if we want to label the Y axis to indicate what it is referring to? Just prior to the plt.plot() code, Type: *plt.ylabel("Temperature")* 5. As an exercise, try labeling the X axis with “Months” as well.   We now have seen how to use pyplot to plot our data. In the next lesson we will start using a library built on top of MatPlotLib called Seaborn. | The learner will load data from a Dataframe into a Pyplot graph. | How do you specify the width of a pyplot? |
| Task 3 | L03 | Welcome back! We’ve seen how to create a plot using pyplot, now we’ll use a larger data set to visualize more interesting patterns using Seaborn.  <SLIDE>  Our goal in this lesson is to learn about how to start using Seaborn to visualize data using a scatterplot graph more easily. We will also see how to check for any possible correlation between two variables through visualization.  Seaborn is built on top of MatPlotlib, so it includes all of the underlying functionality, but makes graphing data more pleasant.   1. Open task3.py in Visual Studio Code.   Include all of the libraries from the last task. Then add : *import seaborn as sns*   1. This time, we will read from a larger file called tempYearly.csv. It includes yearly average rainfall and temperature for a certain region. We first need to read the data, which is in a CSV file: *df = pd.read\_csv(r'tempYearly.csv')* 2. We note the names of the columns in the CSV file, so we can access them to plot the data:  #Temperature, Year and Rainfall 3. Next, we plot Year on the X axis and Temperature on the Y axis:   # 3 Scatter Plot  *sns.scatterplot(x='Year', y='Temperature', data=df)*   1. Next, use the show function to show the plot. *plt.show()* 2. Notice the size of the window is too small to see all of the data. We can use the rc parameters from MatPlotLib to control the width and height. Just before the scatterplot line of code, type   *sns.set(rc={'figure.figsize':(12,6)})*   1. Run the code again, and you’ll see the issue has been fixed! 2. Go ahead and try running a scatterplot on Year and Rainfall this time. Do you see a similar plot? 3. Finally, let us plot Rainfall on the X axis and Temperature on the Y axis: *sns.scatterplot(x='Rainfall', y='Temperature', data=df)*   *plt.show()*   1. It looks as though there is trend in this data. Let us now run a regression line through the scatter and see if there is any possible correlation between rainfall and temperature. Replace the previous scatterplot code with: sns.regplot(x="Rainfall", y="Temperature", data=df) 2. It looks like there is a positive correlation…(by the way, this doesn’t mean cause and effect).   In this section, we learned how to use seaborne to visualize and analyze the data.  In the next section, we will use another data set to create a heatmap. | The learner will use Seaborn to create a scatterplot graph. | What is the proper way to create a scatterplot using Seaborn? |
| Task 4 | L04 | <Slide> Welcome back! In the previous lesson we learned how to use Seaborne to visualize and analyze the data.  In this section we will use a heatmap graph to determine when a high number of births occurred in an unnamed region.  Remember, you can pause the recording at any time to analyze and type in the code, run the code, and make corrections..   * + - 1. Open the file called task4.py. Import the same libraries as we did in task 3.       2. Next, let's take a look at the data in the file birthYearly.csv. It contains column headings month, year, and births. We need to note these so we can plot the appropriate data.       3. A heatmap will visually tell us when the greatest number of birts occurred without having to sift through the numbers themselves.       4. First, let us read and the data using the pandas library function again, and print it as well:   *data = pd.read\_csv(r'./birthYearly.csv')*  *print(data)*   1. Next, let us try to set up a heatmap. The data is the data we just read in. annot is set to true so we can see the birth numbers in the map. Fmt=d means the numbers are whole number decimals.   *sns.heatmap(data, annot=True, fmt="d”)*   1. Next, try to run the code. It won’t run because it is trying to read characters as numbers due to the format of the data. We need to set pivot points to re-arrange the data so it processes the numbers correctly. Remove the previous line of code and type the following to add pivots:   *dataP = data.pivot("month", "year", "births")*  *print(dataP)*   1. Run the code. Do you see how the data has been re-arranged into a matrix? Next, we create the heatmap, this time using the new format. *sns.heatmap(dataP, annot=True, fmt="d"* ) 2. Now, run the code again. You can see the color code bar on the right side. The lighter the box, the higher the number. Conversely, the darker colors mean low numbers.   In this lesson, we discovered how to use Python Pandas to create pivots to format the data properly for Seaborn’s heatmap. We learned that we can quickly visualize quantities of data using the heatmap..  In the next section, we will learn how to visualize certain parts of data within a data set using Seaborn and jointplot. | The learner will Create a heatmap to show correlations | How do you determine the correlation of a data set using Seaborn? |
| Task 5 | L05 | <Slide>  Welcome back! In the previous module, We learned that we can quickly visualize quantities of data using the heatmap.  In this section you will learn how to visualize parts of the data set by selecting parts of the DataFrame using Pandas along with seaborn’s jointplot. With jointplot, you can visualize the distribution behind a scatter plot. Remember, you can pause the recording at any time to analyze and type in the code, run the code, and make corrections..   1. Open the file called task5.py. Import the same libraries you did in the previous task. 2. Next, read in yearly rainfall and temperature data using pandas.   *data = pd.read\_csv(r'./tempYearly.csv')*   1. With Pandas, you can select a particular column from the DataFrame by selecting its name. We will pass just the Temperature and Rainfall only. Type:   *sns.distplot(data[['Rainfall','Temperature']])*  *plt.show()*   The distribution plot shows the likelihood of a value falling in a certain range (the blue area here).   1. Next, take the we will use a jointplot to look at the scatter plot of rainfall vs temperature and also the distribution.  *sns.jointplot("Rainfall","Temperature",data=data)*   *plt.show()*   1. Run this. Notice how you can drag your mouse around the distribution regions surrounding the scatter plot and see the x(rainfall) and y(temperature) values? 2. Finally, add kind=”hex” to the parameters of the jointplot function call as in: *sns.jointplot("Rainfall","Temperature",data=data, kind="hex")* 3. There are many more options when it comes to visualizing data. For more information visit <https://seaborn.pydata.org/>       Congratulations on finishing the course, I hope you enjoy learning about Visualizing Data in Python! | The learner will create a jointplot to show distribution of a dataset. |  |
|  |  |  |  |  |