**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **21st July 2020** | **Name:** | **Mamatha.m** |
| **Course:** | **Python** | **USN:** | **4AL16EC035** |
| **Topic:** | * **Pythonic** | **Semester & Section:** | **6 & B** |
| **GitHub Repository:** | **Mamatha-m** |  |  |

|  |
| --- |
| **SESSION DETAILS**  **Session images**        **Report:**  **Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.**  **Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.** |

**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **21st July 2020** | **Name:** | **Mamatha.m** |
| **Course:** | **coursera** | **USN:** | **4AL16ec035** |
| **Topic:** | **Basic statistics** | **Semester & Section:** | **6 & B** |
| **GitHub Repository:** | **Mamatha-m** |  |  |

|  |
| --- |
| **SESSION DETAILS**  **Session images**  **nbnb.JPG**  **Report:**  **we'll talk about how we can display the correlation between two variables using tables and graphs. First we'll look at categorical variables and discuss contingency tables. In a next step we look at how we can best display the relationship between two quantitative variables. Here we'll introduce the scatterplot.**  **In the second video we'll discuss the Pearson's r - one of the most frequently used measures of correlation. It is an appropriate measure if the variables under analysis are measured on a quantitative level and if they are linearly related to each other. The Pearson's r expresses the direction and strength of the correlation. You'll learn how to interpret the Pearson's r and how to compute it yourself.**  **Regression analysis is one of the most frequently employed statistical methods. In the next three videos we'll discuss the basics of regression analysis. In the first video we'll explain how we can find the regression line (the line that best represents the linear correlation between two quantitative variables in a scatterplot). You'll learn that the best fitting line is the line for which the sum of the squared residuals (vertical distances of the cases in your scatterplot to the line) is the smallest. We therefore talk about ordinary least squares (OLS) regression.**  **In the next video we'll show you how we can describe what the regression line looks like. This is very useful because it can help us make predictions about our dependent variable. We can make these predictions by means of the regression equation of which important ingredients are the regression coefficient and the regression slope. In the final part of this video we'll show you how you can also find the regression line by means of two rather simple formulas.**  **The third video in this section focuses on the question how we can assess how well a regression line fits the data under analysis. Here we'll introduce the so-called r-squared. It tells you how much better a regression line predicts your dependent variable than the mean of that variable, and it shows you how much of the variance in your dependent variable is explained by your independent variable.** |