**DAEN 500- DL1 – Data Analytics Fundamentals**

**Summer 2020 Final Examination**

**6/1 – 7/27/2020**

**Final Submission Deadline: NLT 11:59PM (EST). Monday, 27 July 2020**

***Failure to submit ON TIME will result in DAEN COURSE FAILURE***

**Name:** Ferdinand Percentie **GMU G#** G01291432

**Student Signature (Honor Certification): Ferdinand Percentie (Signature attached)**

This exam is **OPEN BOOK/OPEN NOTES**. You may consult any of the course texts, and the various reference materials recommended in the syllabus. ***The exam of course IS NOT “Open Web”,*** especially in that you may NOT utilize expert “help” sites such as Stack Overflow, or other programming help or collaboration sites.

Additionally, you are restricted from discussing the substance of the questions on this exam with any other individual, until after you have submitted your final response for grading. The completed exam -- with your answers embedded in this docx document (add extra pages as necessary) should be submitted following instructions contained in the Final Exam Instructions BB site. If you have any trouble submitting and have extra parts of the answers you have trouble appending to this document, you may simply submit additional pages separately (the exam submission site is set for multiple submissions, just in case). Make certain all are submitted PRIOR TO THE DEADLINE!

**FINAL EXAM PROBLEMS**

**COMPLETE ALL & INSERT ANSWERS BELOW QUESTIONS**

# **Problem 1: Python Programming Problem (15 Points Total)**

* **Design and implement a Python program that is based on the following requirements: a) program will find all numbers which are divisible by 7 but are not a multiple of 5; and b) numbers between 2000 and 3200.**
* **INSERT (cut&paste) your Python code in space below and *then insert a screen shot in space below, showing your successful run and output.***

NOTE of alternative for help: To help test your code, you also may use a Python “programming window” found in the. **Zybooks Section 35 Additional Material**.

#Script: Question1.py

#Author: Ferdinand Percentie

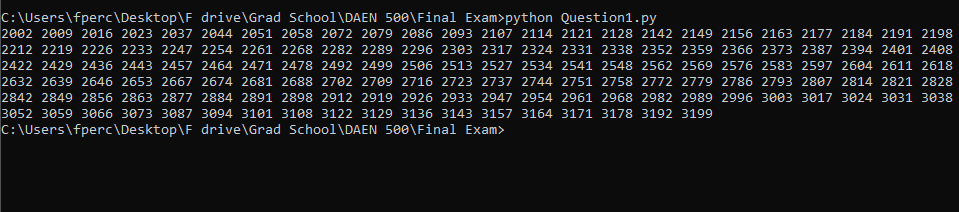
#Date: July 22 2020

'''Design and implement a Python program that is based on the following requirements:

a) program will find all numbers which are divisible by 7 but are not a multiple of 5

b) numbers between 2000 and 3200'''

[print(i, end=' ') for i in range(2000, 3201) if i % 7 == 0 and i % 5 != 0]



# **Problem 2: Python Programming Problem**

# **(15 Points Total)**

* **Design and implement a Python program that is based on the following requirements:**

**a) define a class which has *at least two* methods**

* + **Method 1 – getString: to get a string from console input; and,**
  + **Method 2 - printString: to print the string in upper case.**

**b) demonstrate code works using three different test input strings**

* ***INSERT* *code below* and *INSERT* a screen shot of the program and successfully run output that *includes test input for input strings (test strings must include (a) all upper case, (b) all lower case, and (c) mix of upper and lower case).***

#Script: Question2.py

#Author: Ferdinand

#Date: July 22 2020

class StringPrinter:

'''Class that has two methods: getString and printString'''

def \_\_init\_\_(self):

self.string = ''

def getString(self):

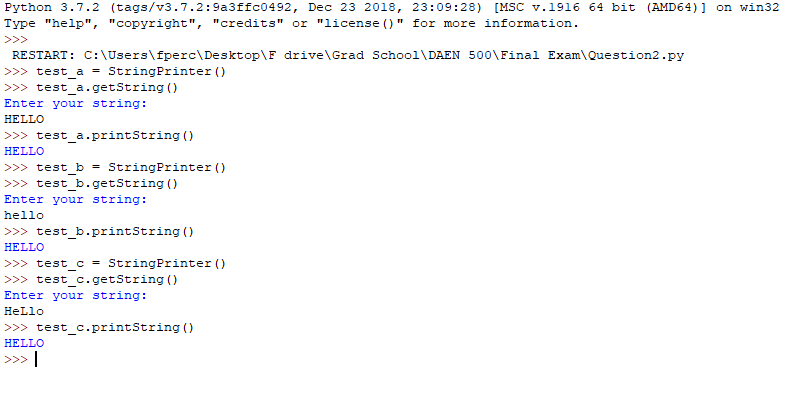
'''Get a string from console input.'''

self.string = input('Enter your string:\n')

def printString(self):

'''To print the string in uppercase.'''

print(self.string.upper())





# **Problem 3: R Programming Problem**

# **(20 Points Total)**

* **Perform the following problems using R:**
  + Create a vector of courses (e.g., MATH 101) you have taken previously. Make sure you have at least 8 courses. Name the vector myCourses
  + Get the length of the vector myCourses
  + Get the first two courses from myCourses
  + Get the 3rd and 4th courses from myCourses
  + Sort myCourses using a method
  + Sort myCourse in the reverse direction
* *INSERT* *code below* and *INSERT* a screen shot of the program and successfully run output.

#Create a vector of courses you have taken previously. At least 8 courses.

myCourses <- c("DAEN 500", "MATH 104", "ECON 221", "PSYC 001",

"URBS 100", "HIST 204", "STAT 104", "FILM 236")

#Length of myCourses

length(myCourses)

#First two courses in myCourses

myCourses[1:2]

#3rd and 4th courses

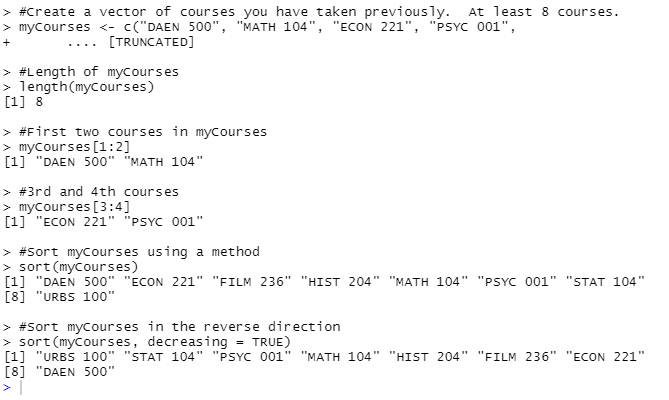
myCourses[3:4]

#Sort myCourses using a method

sort(myCourses)

#Sort myCourses in the reverse direction

sort(myCourses, decreasing = TRUE)





# **Problem 4: Principal Component Analysis**

# **(25 points)**

* **Provide a description of the following:**

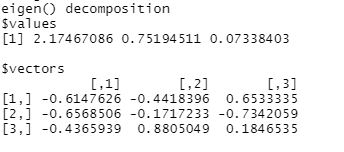
1. What is a component – Provide a description (5 points)
2. Principal Component Analysis – Provide a description.(5 points)
3. **Provide an specific example of Principal Component Analysis(15 points)**
4. A component is a variable formed by combining more than one predictor variables, typically those that are highly correlated or interdependent. Typically correlation values are used to weight the variables in the newly constructed component. (i.e. Z = v1\*x1 + v2\*x2, where Z is the principal component, the v’s are the respective weights and the x’s are the predictor variable)
5. Principal Component Analysis is the process of using these components, combining correlated or interdependent predictor variables, to lessen the number of variables being analyzed and simplify analysis, in order to maximize the data variability along as few components as possible.
6. For example, if you wanted to analyze the prices of a house in a neighborhood, you could combine highly correlated variables into components to simplify the analysis. The square footage of a house may be highly correlated to the number of bedrooms, number of bathrooms, and the property tax bill for the house. Combining these predictor variables into a single component can filter out the noise that comes from having a lot of highly correlated variables and will help from overweighting factors relating to the size of the house in your analysis on price and can help bring out other variables (in this example, quality of the home, safety of the neighborhood, quality of the schools, etc.).

To perform Principal Component Analysis,, you would first arrange a correlation table of the predictor variables. Then, you would convert that correlation table to a correlation matrix, calculate the eigenvalues for that matrix, and use that to calculate eigenvectors for each column. You can then produce the weighted components of the principal components.

Example with Correlation Table using hypothetical variables #Bed, #Bath, Tax:

|  |  |  |  |
| --- | --- | --- | --- |
|  | # Bed | # Bath | Tax |
| # Bed | 1 | 0.9 | 0.3 |
| # Bath | 0.9 | 1 | 0.5 |
| Tax | 0.3 | 0.5 | 1 |

The calculations are performed with the following eigenvalues and vectors:



The principal components produced as a result of this example would result in principal components of::

Z1 = -0.6147626\*X1 -0.4418396\*X2 +0.6533335\*X3

Z2 = -0.6568506\*X1 -0.1717233\*X2 -0.7342059\*X3

Z3 = -0.4365939\*X1 +0.8805049\*X2 +0.1846535\*X3

If these components were too related (i.e. orthogonal to one another when charted), you could simplify further analysis by removing one of those components.

# **Problem 5: Multiple vs. Logistic**

# **(30 points)**

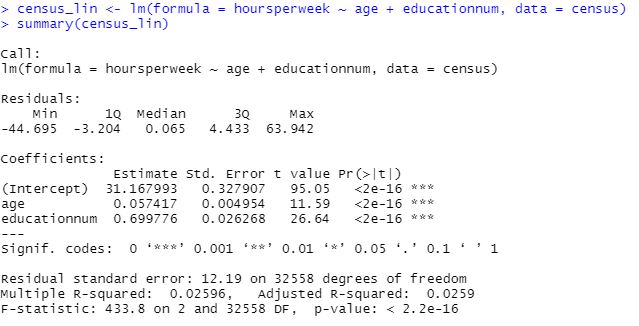
1. Describe: What is difference between Multiple Regression and Logistic Regression? What circumstances might determine which to use? (10 points)
2. Demonstrate: Using any data, and any tool set you’ve learned about, show differences (20 points)

* SUGGESTION: may be solved using RapidMiner, or other toolsets, BOTH TO ANALYZE AND TO VISUALIZE REGRESSION DIFFERENCES..
* Step 1: Perform a quick search of the [UCIS public data archive](https://archive.ics.uci.edu/), a well-curated site which you already have seen as part of your introductory RapidMiner training.
* Step 2: Pick a dataset you find interesting, input dataset into regression tools you’ve chosen.
* Step 3: Run regression, .and use visualizations to demonstrate the conceptual answers you provided for 5.(a).

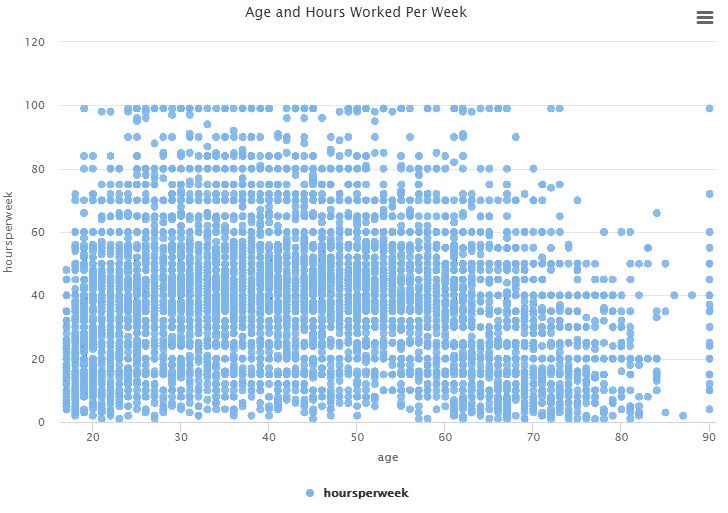
a) Multiple Regression is the tool used to determine the linear relationship between one response variable and one or more predictor variables. Depending on whether the regression equation is representing a sample or population, the equation (Y=β0+β1X1+β2X2+…) may have an added error term (e) if the data comes from a sample.

Logistic Regression is the tool used to model a regression where the response variable has a binary outcome. Whereas multiple regression would fit a straight line to the data, the logisitic regression model fits a “S” -shaped curve and is represented as an exponential function divided by another exponential function. (E(Y)=π(X)=(e^(β0+β1X))/(1+e^(β0+β1X))). Like the multiple regression function, the logistic regression may have an additional error term (e), if the data come from a sample.

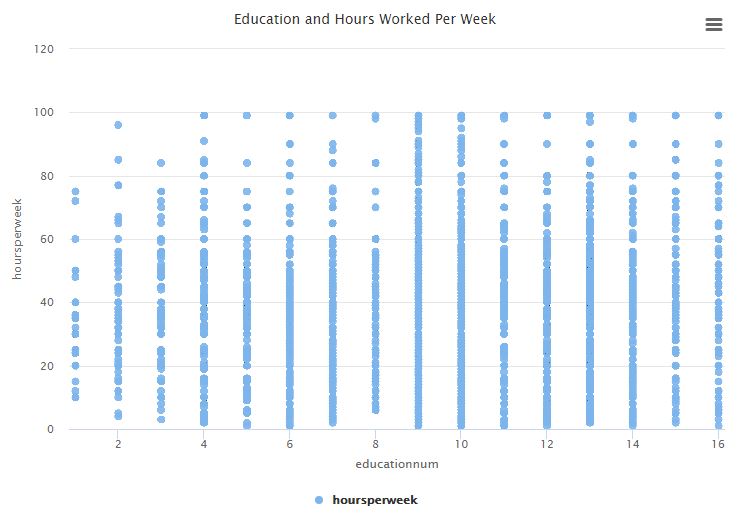
b) Using the adults data set (<https://archive.ics.uci.edu/ml/datasets/Adult>) from UCIS, I've used the hoursperweek column as the response variable and age and educationnum (a ranking of education levels) columns as the predictor variables. For multiple linear regression, I have to use multiple linear regression formula: Y=β0+β1X1+β2X2... The coefficients in the multiple regression model suggest that both predictor variables have somewhat of a positive effect on the hours worked per week when these variables are modeled together. The scatter plots in the graphs below would be used to apply the linear regression line based on the formula above (however, the chart made in RapidMiner refused to produce a regression line for the number of rows of data in the adult data set).

****

**Plot of Age and Hours Worked Per Week**

****

**Plot of Education and Hours Worked Per Week**

****

**Logistic Regression**

Logistic regression however is best used for data where the outcome is a binary response. In the adult data set, we have a binary response variable in the “Over 50K” column (whether or not this row is predictive of an income above $50k). I convert it into a binary variable (0, 1) and use it as the response variable and use hoursperweek as the predictor variable. The regression summary below displays the regression coefficient of 0.04645 for age which suggests that hours per week has a slight effect on the chances of having an income over 50k when the two variables are modeled together. The graph at the bottom of the page is an example of the data, which would be fitted with a S-shaped curve (no example in the Zybooks or Lynda materials) based on the formula: E(Y)=π(X)=(e^(β0+β1X))/(1+e^(β0+β1X)), where B0 and B1 are our coefficient estimates below.

.

