Numpy Assignment

1.Create a numpy array with 10 elements of the shape(10,1) using np.random and find out the mean of the elements using basic numpy functions.

2.Create a numpy array with 20 elements of the shape(20,1) using np.random find the variance and standard deviation of the elements.

3.Create a numpy array A of shape(10,20) and B of shape (20,25) using np.random. Print the matrix which is the matrix multiplication of A and B. The shape of the new matrix should be (10,25). Using basic numpy math functions only find the sum of all the elements of the new matrix.

4.Create a numpy array A of shape(10,1).Using the basic operations of the numpy array generate an array of shape(10,1) such that each element is the following function applied on each element of A.

#f(x)=1 / (1 + exp(-x))

#Apply this function to each element of A and print the new array holding the value the function returns

#Example:

#a=[a1,a2,a3]

#n(new array to be printed)=[f(a1), f(a2), f(a3)]

5. Generate a matrix of 20 random values and find its cumulative sum

6. x = np.array([[1., 2., 3.], [4., 5., 6.]]) y = np.array([[6., 23.], [-1, 7], [8, 9]])

Find the dot product of the above two matrix

7. Following is the input NumPy array delete column two and insert following new column in its place.

sampleArray = numpy.array([[34,43,73],[82,22,12],[53,94,66]])

newColumn = numpy.array([[10,10,10]])

8. names = np.array(['Bob', 'Joe', 'Will', 'Bob', 'Will', 'Joe', 'Joe'])

Find the unique names and sort them out!

9. Create two random arrays of range 12 and make an array with the maximum values between each element of the two arrays

10. names = np.array(['Bob', 'Joe', 'Will', 'Bob', 'Will', 'Joe', 'Joe']) data = np.random.randn(7, 4)

Find all the values from array data where the values from array names are not equal to Will

11. How to find the memory size of any array

12. How to get the documentation of the numpy add function from the command line?

13. Create a null vector of size 10 but the fifth value which is 1

14. Create a vector with values ranging from 10 to 49

15. Reverse a vector (first element becomes last)

16. Create a 3x3 matrix with values ranging from 0 to 8

17. Find indices of non-zero elements from [1,2,0,0,4,0]

18. Create a 3x3 identity matrix

19. Create a 3x3x3 array with random values

20. Create a 10x10 array with random values and find the minimum and maximum values

21. Create a random vector of size 30 and find the mean value

22. Create a 2d array with 1 on the border and 0 inside

23. How to add a border (filled with 0's) around an existing array? (★☆☆) ####

24. What is the result of the following expression?

*0 *np.nan*

np.nan == np.nan

np.inf > np.nan

np.nan - np.nan

*np.nan in set([np.nan])0.3 == 3 * 0.1*

25. Create a 5x5 matrix with values 1,2,3,4 just below the diagonal

26. Create a 8x8 matrix and fill it with a checkerboard pattern

27. Consider a (6,7,8) shape array, what is the index (x,y,z) of the 100th element?

28. Create a checkerboard 8x8 matrix using the tile function

29. Normalize a 5x5 random matrix

30. Create a custom dtype that describes a color as four unsigned bytes (RGBA)

31. Multiply a 5x3 matrix by a 3x2 matrix (real matrix product)

32. Given a 1D array, negate all elements which are between 3 and 8, in place.

33. What is the output of the following script?

print(sum(range(5),-1))

*from numpy import **

print(sum(range(5),-1))

34. Consider an integer vector Z, which of these expressions are legal?

*Z**Z*

2 << Z >> 2

Z <- Z

*1j*Z*

Z/1/1

Z<Z>Z

35. What are the result of the following expressions?

pythonnp.array(0) / np.array(0)

np.array(0) // np.array(0)

np.array([np.nan]).astype(int).astype(float)

36. How to round away from zero a float array ?

37. How to find common values between two arrays?

38. How to ignore all numpy warnings (not recommended)?

39. Is the following expressions true?

np.sqrt(-1) == np.emath.sqrt(-1)

40. How to get the dates of yesterday, today and tomorrow?

41. How to get all the dates corresponding to the month of July 2016?

42. How to compute ((A+B)*(-A/2)) in place (without copy)?

43. Extract the integer part of a random array of positive numbers using 4 different methods

44. Create a 5x5 matrix with row values ranging from 0 to 4

45. Consider a generator function that generates 10 integers and use it to build an array

46. Create a vector of size 10 with values ranging from 0 to 1, both excluded

47. Create a random vector of size 10 and sort it

48. How to sum a small array faster than np.sum?

49. Consider two random array A and B, check if they are equal

50. Make an array immutable (read-only)

51. Consider a random 10x2 matrix representing cartesian coordinates, convert them to polar coordinates

52. Create random vector of size 10 and replace the maximum value by 0

53. Create a structured array with `x` and `y` coordinates covering the [0,1]x[0,1] area

54. Given two arrays, X and Y, construct the Cauchy matrix C (Cij =1/(xi - yj))

55. Print the minimum and maximum representable value for each numpy scalar type

56. How to print all the values of an array?

57. How to find the closest value (to a given scalar) in a vector?

58. Create a structured array representing a position (x,y) and a color (r,g,b)

59. Consider a random vector with shape (100,2) representing coordinates, find point by point distances

60. How to convert a float (32 bits) array into an integer (32 bits) in place?

61. How to read the following file?

1, 2, 3, 4, 5

6, , , 7, 8

, , 9,10,11

62. What is the equivalent of enumerate for numpy arrays?

63. Generate a generic 2D Gaussian-like array

64. How to randomly place p elements in a 2D array?

65. Subtract the mean of each row of a matrix

66. How to sort an array by the nth column?

66. How to tell if a given 2D array has null columns?

67. Find the nearest value from a given value in an array

68. Considering two arrays with shape (1,3) and (3,1), how to compute their sum using an iterator?

69. Create an array class that has a name attribute

70. Consider a given vector, how to add 1 to each element indexed by a second vector (be careful with repeated indices)?

71. How to accumulate elements of a vector (X) to an array (F) based on an index list (I)?

72. Considering a (w,h,3) image of (dtype=ubyte), compute the number of unique colors

73. Considering a four dimensions array, how to get sum over the last two axis at once?

74. Considering a one-dimensional vector D, how to compute means of subsets of D using a vector S of same size describing subset indices?

75. How to get the diagonal of a dot product?

76. Consider the vector [1, 2, 3, 4, 5], how to build a new vector with 3 consecutive zeros interleaved between each value?

77. Consider an array of dimension (5,5,3), how to mulitply it by an array with dimensions (5,5)? 78. Consider a set of 10 triplets describing 10 triangles (with shared vertices), find the set of unique line segments composing all the triangles

79. Given a sorted array C that corresponds to a bincount, how to produce an array A such that np.bincount(A) == C?

80. How to compute averages using a sliding window over an array?

81. Consider a one-dimensional array Z, build a two-dimensional array whose first row is (Z[0],Z[1],Z[2]) and each subsequent row is shifted by 1 (last row should be (Z[-3],Z[-2],Z[-1])

82. How to negate a boolean, or to change the sign of a float inplace?

83. Consider 2 sets of points P0,P1 describing lines (2d) and a point p, how to compute distance from p to each line i (P0[i],P1[i])?

84. Consider 2 sets of points P0,P1 describing lines (2d) and a set of points P, how to compute distance from each point j (P[j]) to each line i (P0[i],P1[i])?

85. Consider an arbitrary array, write a function that extract a subpart with a fixed shape and centered on a given element (pad with a `fill` value when necessary)

86. Consider an array Z = [1,2,3,4,5,6,7,8,9,10,11,12,13,14], how to generate an array R = [[1,2,3,4], [2,3,4,5], [3,4,5,6], ..., [11,12,13,14]]?

87. Compute a matrix rank

88. How to find the most frequent value in an array?

89. Extract all the contiguous 3x3 blocks from a random 10x10 matrix

90. Create a 2D array subclass such that Z[i,j] == Z[j,i]

91. Consider a set of p matrices wich shape (n,n) and a set of p vectors with shape (n,1). How to compute the sum of of the p matrix products at once? (result has shape (n,1))

92. Consider a 16x16 array, how to get the block-sum (block size is 4x4)?

93. How to implement the Game of Life using numpy arrays?

94. How to get the n largest values of an array

95. Given an arbitrary number of vectors, build the cartesian product (every combinations of every item)

96. How to create a record array from a regular array?

97. Consider a large vector Z, compute Z to the power of 3 using 3 different methods

98. Consider two arrays A and B of shape (8,3) and (2,2). How to find rows of A that contain elements of each row of B regardless of the order of the elements in B?

99. Considering a 10x3 matrix, extract rows with unequal values (e.g. [2,2,3])

100. Convert a vector of ints into a matrix binary representation

101. Given a two dimensional array, how to extract unique rows?

102. Considering 2 vectors A & B, write the einsum equivalent of inner, outer, sum, and mul function

103. Considering a path described by two vectors (X,Y), how to sample it using equidistant samples

104. Given an integer n and a 2D array X, select from X the rows which can be interpreted as draws from a multinomial distribution with n degrees, i.e., the rows which only contain integers and which sum to n.

105. Compute bootstrapped 95% confidence intervals for the mean of a 1D array X (i.e., resample the elements of an array with replacement N times, compute the mean of each sample, and then compute percentiles over the means).

106. How to swap two rows of an array?

FOR PROBLEM BELOW YOU CAN MAKE YOUR OWN DATASET (each data set should have at least 100 records)

107. Analyzing student grades to identify areas where students are struggling. Dataset: A CSV file containing grades for all students in a particular course.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the mean and standard deviation for the grades in the course.
- Compute the median and interquartile range for the grades in the course.
- Identify the top-performing and bottom-performing students in the course based on their grades.

108. Analyzing research publication data to identify trends in faculty research areas. Dataset: A CSV file containing information about research publications by faculty members in the past year.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total number of publications for each faculty member.
- Compute the mean and standard deviation for the number of publications across all faculty members.
- Identify the faculty member(s) with the highest number of publications.

109. Analyzing student enrollment data to identify trends in course enrollment. Dataset: A CSV file containing enrollment data for all courses offered in the current semester.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total number of students enrolled in each course.
- Compute the mean and standard deviation for the number of students enrolled across all courses.
- Identify the course(s) with the highest and lowest number of students enrolled.

110. Analyzing survey responses to identify trends in student satisfaction with campus services. Dataset: A CSV file containing survey responses from all students in the university.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the average satisfaction score for each campus service.
- Identify the service(s) with the lowest satisfaction score.
- Compute the standard deviation for the satisfaction scores for each campus service.

111. Analyzing student course enrollment data to identify prerequisites for popular courses. Dataset: A CSV file containing information about the prerequisites for all courses offered in the current semester.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total number of students enrolled in each course.
- Identify the course(s) with the highest number of students enrolled.
- Identify the prerequisite course(s) for the popular course(s).

112. Analyzing stock market data to identify trends in stock prices and trading volumes. Dataset: A CSV file containing daily stock prices and trading volumes for a particular company.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the daily percentage change in the stock price.
- Compute the average daily trading volume.
- Identify the days with the highest and lowest trading volumes.

113. Analyzing household income data to identify trends in income inequality. Dataset: A CSV file containing household income data for a particular region.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the mean and median household incomes.
- Compute the Gini coefficient for the household income distribution.

- Identify the percentage of households living below the poverty line.

114. Analyzing trade data to identify trends in exports and imports. Dataset: A CSV file containing monthly trade data for a particular country.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the monthly trade balance (i.e., exports minus imports).
- Compute the average monthly exports and imports.
- Identify the months with the highest and lowest trade balances.

115. Analyzing demographic data to identify trends in population growth and migration. Dataset: A CSV file containing demographic data for a particular region, including population counts by age group and migration flows.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the percentage change in population size over a given time period.
- Compute the net migration for the region.
- Identify the age group(s) with the highest and lowest population growth rates.

116. Analyzing economic data to identify trends in inflation and unemployment rates. Dataset: A CSV file containing monthly economic data for a particular country, including consumer price index (CPI) and unemployment rate.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the monthly inflation rate.
- Compute the average unemployment rate.
- Identify the months with the highest and lowest inflation rates.

117. Analyzing customer purchase data to identify trends in customer behavior. Dataset: A CSV file containing customer purchase data for a particular e-commerce website.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total number of purchases made by each customer.
- Compute the average purchase value for each customer.
- Identify the top-spending customers.

118. Analyzing survey data to identify trends in customer satisfaction. Dataset: A CSV file containing survey data from customers of a particular company.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the average satisfaction score for each question in the survey.
- Identify the question(s) with the lowest satisfaction score.
- Compute the correlation between customer satisfaction and repeat business.

119. Analyzing website traffic data to identify trends in user behavior. Dataset: A CSV file containing website traffic data for a particular website.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total number of unique visitors to the website.
- Compute the average time spent on the website by each visitor.
- Identify the pages with the highest and lowest bounce rates.

120. Problem: Analyzing customer feedback data to identify trends in product satisfaction. Dataset: A CSV file containing customer feedback data for a particular product, including ratings and comments.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the average rating for the product.
- Identify the positive and negative aspects of the product based on customer comments.
- Compute the correlation between product ratings and sales.

121. Analyzing social media data to identify trends in customer sentiment. Dataset: A CSV file containing social media data related to a particular brand, including text and sentiment scores.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the average sentiment score for all social media posts related to the brand.
- Identify the most common positive and negative sentiments expressed in social media posts.
- Compute the correlation between social media sentiment and customer loyalty.

122. Analyzing fitness tracker data to identify trends in physical activity. Dataset: A CSV file containing physical activity data recorded by a fitness tracker.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total number of steps taken each day.
- Compute the average number of steps taken per day.
- Identify the days with the highest and lowest number of steps taken.

123. Analyzing sleep data to identify trends in sleep patterns. Dataset: A CSV file containing sleep data recorded by a sleep tracker.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total time slept each night.
- Compute the average time slept per night.
- Identify the nights with the longest and shortest sleep durations.

124. Analyzing grocery purchase data to identify trends in spending. Dataset: A CSV file containing grocery purchase data for a particular household.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total amount spent on groceries each week.
- Compute the average amount spent on groceries each week.
- Identify the weeks with the highest and lowest grocery expenditures.

125. Analyzing travel data to identify trends in transportation modes. Dataset: A CSV file containing travel data for a particular trip, including modes of transportation and durations.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total time spent in each mode of transportation.
- Compute the average time spent in each mode of transportation.
- Identify the modes of transportation with the longest and shortest durations.

126. Analyzing weather data to identify trends in temperature. Dataset: A CSV file containing temperature data for a particular location.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the daily average temperature.
- Compute the average temperature for each month.
- Identify the months with the highest and lowest average temperatures.

127. Analyzing financial data to identify trends in expenses. Dataset: A CSV file containing monthly financial data for a particular household.

Exercise:

- Load the CSV file into a NumPy array.

- Compute the total expenses for each month.
- Compute the average monthly expenses.
- Identify the months with the highest and lowest expenses.

128. Analyzing academic data to identify trends in grades. Dataset: A CSV file containing grade data for a particular student.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the average grade for each course.
- Identify the course(s) with the highest and lowest average grades.
- Compute the student's overall GPA.

129. Analyzing music listening data to identify trends in listening habits. Dataset: A CSV file containing music listening data for a particular individual, including artists and genres listened to.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total listening time for each artist.
- Compute the total listening time for each genre.
- Identify the artist(s) and genre(s) with the most listening time.

130. Analyzing commuting data to identify trends in travel time. Dataset: A CSV file containing commuting data for a particular individual.

Exercise:

- Load the CSV file into a NumPy array.
- Compute the total time spent commuting each week.
- Compute the average time spent commuting each day.
- Identify the days with the longest and shortest commutes.

131. Write a Python program that uses NumPy to perform linear regression on a set of data points. (use your generated data)

132. Write a Python program that loads a CSV file containing two sets of data and computes the correlation between the two sets using NumPy. (use your generated data)

133. Write a Python program that loads a CSV file containing time series data and computes a moving average of the data using NumPy. (use your generated data)

134. Write a Python program that loads a CSV file containing a set of data points and computes the covariance matrix of the data using NumPy. (use your generated data)

135. Write a Python program that creates a NumPy array and performs basic mathematical operations such as exponentiation, logarithmic functions, and trigonometric functions. (use your generated data)

ADVANCE LEVEL (OPTIONAL) *** BONUS SCORE******

1. Write a Python program that creates a NumPy array and uses NumPy to perform basic statistical hypothesis testing such as t-tests, ANOVA, and chi-square tests.
2. Write a Python program that creates a NumPy array and uses NumPy to perform basic Fourier analysis to decompose a signal into its frequency components.
3. Write a Python program that loads a CSV file containing a set of data and uses NumPy to perform basic time series analysis such as autocorrelation, moving average, and seasonality analysis.
4. Write a Python program that loads a CSV file containing a set of data and uses NumPy to perform basic machine learning tasks such as classification, clustering, or regression.
5. Write a Python program that loads a CSV file containing a set of data and uses NumPy to perform time series forecasting on the data.