**Activities: NumPy and Pandas**

The activities on this page allow you to demonstrate your ability to:

* Create, manipulate, and analyze data using [NumPy](https://academy.engagelms.com/mod/book/view.php?id=185672) arrays and [Pandas](https://academy.engagelms.com/mod/book/view.php?id=185673) dataframes.
* Perform mathematical operations, aggregation, and [sorting](https://academy.engagelms.com/mod/page/view.php?id=185640) on data.

**Activity 1**

Create a [NumPy](https://academy.engagelms.com/mod/book/view.php?id=185672) array with 20 random integers between 1 and 100.

Compute and display only the mean, median, and standard deviation of the values in the array.

|  |
| --- |
| import numpy as np  # Create an array with 20 random integers between 1 and 100  integers = np.random.randint(1, 101, size=20)  # Compute the mean, median, and standard deviation  mean\_value = np.mean(integers)  median\_value = np.median(integers)  std\_deviation = np.std(integers)  # Display the results  print("Array:", integers)  print("Mean:", mean\_value)  print("Median:", median\_value)  print("Standard Deviation:", std\_deviation)  Output:  Array: [  8  83  17   1  98   9  94  28  91  71   6   1  21  93  28  18  64  52    26 100]  Mean: 45.45  Median: 28.0  Standard Deviation: 36.24013658914657 |

**Activity 2**

Create a 5x5 [NumPy](https://academy.engagelms.com/mod/book/view.php?id=185672) array with random integers between 1 and 50.

Calculate and display the sum of each row and each column.

|  |
| --- |
| import numpy as np  # Create a 5x5 array with random integers between 1 and 50  array\_5x5 = np.random.randint(1, 51, size=(5, 5))  # Calculate the sum of each row and column  row\_sums = np.sum(array\_5x5, axis=1)  column\_sums = np.sum(array\_5x5, axis=0)  # Display the array and the results  print("5x5 Array: \n", array\_5x5)  print("Sum of each row: ", row\_sums)  print("Sum of each column: ", column\_sums)  Output:  5x5 Array:   [[ 2  6 41 10 24]   [42 25 27 11 32]   [ 6 30 17 47 31]   [15 27 23 21 36]   [ 1 40 25 31 41]]  Sum of each row:  [ 83 137 131 122 138]  Sum of each column:  [ 66 128 133 120 164] |

**Activity 3**

Create a [Pandas](https://academy.engagelms.com/mod/book/view.php?id=185673) DataFrame from the following dictionary:

data = {

"Product": ["Apple", "Banana", "Orange", "Grape", "Watermelon"],

"Price": [1.5, 0.75, 1, 2, 4],

"Quantity": [10, 20, 30, 5, 8]

}

Calculate and display the total value of each product (Price \* Quantity).

|  |
| --- |
| import pandas as pd  # Create a dictionary  data = {      "Product": ["Apple", "Banana", "Orange", "Grape", "Watermelon"],      "Price": [1.5, 0.75, 1, 2, 4],      "Quantity": [10, 20, 30, 5, 8]  }  # Create a DataFrame  df = pd.DataFrame(data)  # Calculate the total value of each product (Total value = Price \* Quantity)  df['Total Value'] = df['Price'] \* df['Quantity']  print(df)  Output:        Product  Price  Quantity  Total Value  0       Apple   1.50        10         15.0  1      Banana   0.75        20         15.0  2      Orange   1.00        30         30.0  3       Grape   2.00         5         10.0  4  Watermelon   4.00         8         32.0 |

**Activity 4**

Load the dataset provided below into a [Pandas](https://academy.engagelms.com/mod/book/view.php?id=185673) DataFrame and perform the following tasks:

* Calculate and display the average age of all passengers.
* Calculate and display the number of passengers who survived.
* Calculate and display the percentage of passengers who survived.

import [pandas](https://academy.engagelms.com/mod/book/view.php?id=185673) as pd

url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"

titanic = pd.read\_csv(url)

|  |
| --- |
| import pandas as pd  # Load the dataset  url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"  titanic = pd.read\_csv(url)  # Calculate the average age of all passengers  average\_age = titanic['Age'].mean()  # Calculate the number of passengers who survived  survived\_count = titanic['Survived'].sum()  # Calculate the percentage of passengers who survived  total\_passengers = len(titanic)  survived\_percentage = (survived\_count / total\_passengers) \* 100  print("Average age of all passengers:", average\_age)  print("Number of passengers who survived:", survived\_count)  print("Percentage of passengers who survived:", survived\_percentage)  Output:  Average age of all passengers: 29.69911764705882  Number of passengers who survived: 342  Percentage of passengers who survived: 38.38383838383838 |

**Activity 5**

Using the Titanic dataset from the previous activity, calculate and display the following:

* The average age of the passengers who survived.
* The average age of the passengers who did not survive.
* The number of passengers who survived, grouped by gender.
* The number of passengers who did not survive, grouped by gender.

|  |
| --- |
| import pandas as pd  # Load the dataset  url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"  titanic = pd.read\_csv(url)  # Calculate the average age of the passengers who survived  average\_age\_survived = titanic[titanic['Survived'] == 1]['Age'].mean()  # Calculate the average age of the passengers who did not survive  average\_age\_not\_survived = titanic[titanic['Survived'] == 0]['Age'].mean()  # Calculate the number of passengers who survived, grouped by gender  survived\_by\_gender = titanic[titanic['Survived'] == 1]['Sex'].value\_counts()  # Calculate the number of passengers who did not survive, grouped by gender  not\_survived\_by\_gender = titanic[titanic['Survived'] == 0]['Sex'].value\_counts()  # Display the results  print("Average age of passengers who survived:", average\_age\_survived)  print("Average age of passengers who did not survive:", average\_age\_not\_survived)  print("Number of passengers who survived, grouped by gender: \n", survived\_by\_gender)  print("Number of passengers who did not survive, grouped by gender: \n", not\_survived\_by\_gender)  Output:  Average age of passengers who survived: 28.343689655172415  Average age of passengers who did not survive: 30.62617924528302  Number of passengers who survived, grouped by gender:   Sex  female    233  male      109  Name: count, dtype: int64  Number of passengers who did not survive, grouped by gender:   Sex  male      468  female     81  Name: count, dtype: int64 |

**Activity 6**

Using the same Titanic dataset from the earlier activities, create a new DataFrame that includes only the following columns:

* "PassengerId"
* "Survived"
* "Pclass"
* "Name"
* "Sex"
* "Age"
* "Fare"

Sort this DataFrame by "Fare" in descending order and display the first 10 rows.

|  |
| --- |
| import pandas as pd  # Load the dataset  url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"  titanic = pd.read\_csv(url)  # Create a new DataFrame  selected\_columns = ["PassengerId", "Survived", "Pclass", "Name", "Sex", "Age", "Fare"]  titanic\_selected = titanic[selected\_columns]  # Sort the DataFrame by "Fare" in descending order  titanic\_sorted = titanic\_selected.sort\_values(by="Fare", ascending=False)  print(titanic\_sorted.head(10)) |

**Activity 7**

Create a [Pandas](https://academy.engagelms.com/mod/book/view.php?id=185673) DataFrame with the following data:

grades = {

"Student": ["Alice", "Bob", "Charlie", "David", "Eve"],

"Math": [90, 80, 85, 95, 75],

"Science": [85, 95, 80, 75, 90],

"English": [80, 85, 90, 75, 95]

}

Calculate and display the average grade for each student and the average grade for each subject.

|  |
| --- |
| import pandas as pd  # Create a dictionary  grades = {      "Student": ["Alice", "Bob", "Charlie", "David", "Eve"],      "Math": [90, 80, 85, 95, 75],      "Science": [85, 95, 80, 75, 90],      "English": [80, 85, 90, 75, 95]  }  # Create a DataFrame  grades\_df = pd.DataFrame(grades)  # Calculate the average grade for each student  grades\_df['Average\_Grade'] = grades\_df[['Math', 'Science', 'English']].mean(axis=1)  # Calculate the average grade for each subject  average\_grades\_subject = grades\_df[['Math', 'Science', 'English']].mean()  print("Grades DataFrame with Average Grade for Each Student: \n", grades\_df)  print("\n Average Grade for Each Subject: \n", average\_grades\_subject) |

**Activity 8**

Using the grades DataFrame from the previous activity, create a new column named "Status" with the following criteria:

* If a student's average grade is 90 or above, the status should be "Excellent."
* If a student's average grade is between 80 and 89, the status should be "Good."
* If a student's average grade is below 80, the status should be "Needs Improvement."

Display the updated DataFrame with the new "Status" column.

|  |
| --- |
| import pandas as pd  # Create a dictionary  grades = {      "Student": ["Alice", "Bob", "Charlie", "David", "Eve"],      "Math": [90, 80, 85, 95, 75],      "Science": [85, 95, 80, 75, 90],      "English": [80, 85, 90, 75, 95]  }  grades\_df = pd.DataFrame(grades)  # Calculate the average grade for each student  grades\_df['Average\_Grade'] = grades\_df[['Math', 'Science', 'English']].mean(axis=1)  # Define a function to determine status based on average grade  def determine\_status(average\_grade):      if average\_grade >= 90:          return "Excellent"      elif 80 <= average\_grade < 90:          return "Good"      else:          return "Needs Improvement"  # Apply the function to create the "Status" column  grades\_df['Status'] = grades\_df['Average\_Grade'].apply(determine\_status)  print("Grades DataFrame with Status Column: \n", grades\_df) |