NumPy and DataFrames

(AI notes)

list = [10, 20, 14, 50, 23]

// Create a numpy list (used mostly for easy mathematical calculus)
numpyList = np.array(list)

// Calculate the average from the list
averageNumber = numpyList.mean()

// We use pandas library to display a table (using DataFrame) with 3 columns:

```
import pandas as pd

df_students = pd.DataFrame({'Name': ['Dan', 'Joann', 'Pedro', 'Rosie',
'Ethan', 'Vicky', 'Frederic', 'Jimmie',
                                     'Rhonda', 'Giovanni', 'Francesca',
'Rajab', 'Naiyana', 'Kian', 'Jenny',

'Jakeem','Helena','Ismat','Anila','Skye','Daniel','Aisha'],
                            'StudyHours':student_data[0],
                            'Grade':student_data[1]})

df_students
```

// Use DataFrame's loc method to take the data from a row:

```
df_students.loc[5]
```

// … or the data in a range:

```
df_students.loc[0:5]
```

// it takes rows 0, 1, 2, 3, 4, 5

// Use iloc method to take data in a range (different from loc by not also taking the 5-th row)

```
df_students.iloc[0:5]
```

// it takes rows 0, 1, 2, 3, 4

// Also, we can take data from some specific columns

```
df_students.iloc[0,[1,2]]
```

// it takes the first row with the data from the second and third column

// Use the loc method to find indexed rows based on a filtering expression

```
df_students.loc[df_students['Name']=='Aisha']
```

OR

```
df_students[df_students['Name']=='Aisha']
```

OR

```
df_students[df_students.Name == 'Aisha']
```

OR // with query method

```
df_students.query('Name=="Aisha"')
```

// This is how you take the data from a file using pandas:

```
!wget
https://raw.githubusercontent.com/MicrosoftDocs/mslearn-introduction-to
-machine-learning/main/Data/ml-basics/grades.csv
df_students = pd.read_csv('grades.csv',delimiter=',',header='infer')
df_students.head()
```

// We use isnull method to identify which individual values are null:

```
df_students.isnull()
```

// We see just the rows in which any of the columns are null:

```
df_students[df_students.isnull().any(axis=1)]
```

// We can use fillna method to change the null values for StudyHours with the average number:

```
df_students.StudyHours =
df_students.StudyHours.fillna(df_students.StudyHours.mean())
df_students
```

// We can use dropna method to eliminate the rows where there are any null columns:

```
df_students = df_students.dropna(axis=0, how='any')
df_students
```

// Exploring the data in the DataFrame:

```
# Get the mean study hours using to column name as an index
mean_study = df_students['StudyHours'].mean()

# Get the mean grade using the column name as a property (just to make
the point!)
mean_grade = df_students.Grade.mean()

# Print the mean study hours and mean grade
print('Average weekly study hours: {:.2f}\nAverage grade:
{:.2f}'.format(mean_study, mean_grade))
```

// We can group the rows by columns using groupby:

```
print(df_students.groupby(df_students.Pass).Name.count())
```

// We can sort the DataFrame like this:

```
# Create a DataFrame with the data sorted by Grade (descending)
df_students = df_students.sort_values('Grade', ascending=False)

# Show the DataFrame
df_students
```