2330016056

September 16, 2025

1 GTSC2143 Machine Learning for Business - Tutorial 2

1.1 Activity 1: Pandas Series Operations

[ ]:
```
import pandas as pd

# Create a Series with daily temperatures
temps = pd.Series([22, 25, 19, 21, 24], index=["Mon", "Tue", "Wed", "Thu", "Fri"])
temps
```

[ ]:
```
Mon    22
Tue    25
Wed    19
Thu    21
Fri    24
dtype: int64
```

[ ]:
```
# Access Wednesday's temperature and first 3 days
print("Wednesday:", temps.loc["Wed"])
temps.iloc[:3]
```

```
Wednesday: 19
```

[ ]:
```
Mon    22
Tue    25
Wed    19
dtype: int64
```

[ ]:
```
# Find days with temperature above 22 degrees
temps[temps > 22]
```

[ ]:
```
Tue    25
Fri    24
dtype: int64
```

[ ]:
```
# Convert to Fahrenheit
temps_f = temps * 9/5 + 32
temps_f
```

```
[ ]: Mon    71.6
     Tue    77.0
     Wed    66.2
     Thu    69.8
     Fri    75.2
     dtype: float64
```

```
[ ]: # Mean temperature and hottest day
     print("Mean:", temps.mean())
     print("Hottest day:", temps.idxmax(), "with", temps.max(), "°C")
```

```
Mean: 22.2
Hottest day: Tue with 25 °C
```

1.2 Activity 2: Pandas DataFrame

```
[ ]: student_data = {
         'StudentID': [2001, 2002, 2003, 2004, 2005, 2006],
         'Name': ['Alice Wong', 'Bob Chen', 'Carol Lee', 'David Kim', 'Emma Liu',␣
      ↪'Frank Tan'],
         'Age': [20, 21, 19, 22, 20, 21],
         'Major': ['Computer Science', 'Mathematics', 'Physics', 'Computer Science',␣
      ↪'Mathematics', 'Physics'],
         'Math_Score': [85, 92, 78, 96, 88, 91],
         'English_Score': [78, 85, 92, 82, 91, 86]
     }
     df_students = pd.DataFrame(student_data)
     df_students
```

```
[ ]:    StudentID        Name  Age             Major  Math_Score  English_Score
     0       2001  Alice Wong   20  Computer Science          85             78
     1       2002    Bob Chen   21       Mathematics          92             85
     2       2003   Carol Lee   19           Physics          78             92
     3       2004   David Kim   22  Computer Science          96             82
     4       2005    Emma Liu   20       Mathematics          88             91
     5       2006   Frank Tan   21           Physics          91             86
```

```
[ ]: df_students.shape, df_students.head(3)
```

```
[ ]: ((6, 6),
         StudentID        Name  Age             Major  Math_Score  English_Score
      0       2001  Alice Wong   20  Computer Science          85             78
      1       2002    Bob Chen   21       Mathematics          92             85
      2       2003   Carol Lee   19           Physics          78             92)
```

```
[ ]: df_students[['Math_Score','English_Score']].describe()
[ ]:        Math_Score  English_Score
     count    6.000000       6.000000
     mean    88.333333      85.666667
     std      6.282250       5.316641
     min     78.000000      78.000000
     25%     85.750000      82.750000
     50%     89.500000      85.500000
     75%     91.750000      89.750000
     max     96.000000      92.000000
```

```
[ ]: df_students.dtypes
```

```
[ ]: StudentID         int64
     Name             object
     Age               int64
     Major            object
     Math_Score        int64
     English_Score     int64
     dtype: object
```

1.3 Activity 3: DataFrame Operations

```
[ ]: df_students[['Name', 'Math_Score']]
```

```
[ ]:          Name  Math_Score
     0  Alice Wong          85
     1    Bob Chen          92
     2   Carol Lee          78
     3   David Kim          96
     4    Emma Liu          88
     5   Frank Tan          91
```

```
[ ]: df_students[df_students['Math_Score'] > 85]
```

```
[ ]:    StudentID       Name  Age             Major  Math_Score  English_Score
     1       2002   Bob Chen   21       Mathematics          92             85
     3       2004  David Kim   22  Computer Science          96             82
     4       2005   Emma Liu   20       Mathematics          88             91
     5       2006  Frank Tan   21           Physics          91             86
```

```
[ ]: df_students[df_students['Major'] == 'Computer Science']
```

```
[ ]:    StudentID        Name  Age             Major  Math_Score  English_Score
     0       2001  Alice Wong   20  Computer Science          85             78
     3       2004   David Kim   22  Computer Science          96             82
```

```
[ ]: df_students[df_students['English_Score'] > df_students['Math_Score']]
[ ]:    StudentID       Name  Age        Major  Math_Score  English_Score
     2       2003  Carol Lee   19      Physics          78             92
     4       2005   Emma Liu   20  Mathematics          88             91
```

```
[ ]: df_students['Total_Score'] = df_students['Math_Score'] +␣
      ↪df_students['English_Score']
     def grade_bucket(total):
         if total >= 170:
             return "High"
         elif 150 <= total <= 169:
             return "Medium"
         else:
             return "Low"
     df_students['Grade_Level'] = df_students['Total_Score'].apply(grade_bucket)
     df_students.sort_values(by='Total_Score', ascending=False)
```

```
[ ]:    StudentID        Name  Age             Major  Math_Score  English_Score  \
     4       2005    Emma Liu   20       Mathematics          88             91
     3       2004    David Kim   22  Computer Science          96             82
     1       2002     Bob Chen   21       Mathematics          92             85
     5       2006    Frank Tan   21           Physics          91             86
     2       2003    Carol Lee   19           Physics          78             92
     0       2001   Alice Wong   20  Computer Science          85             78

        Total_Score Grade_Level
     4          179        High
     3          178        High
     1          177        High
     5          177        High
     2          170        High
     0          163      Medium
```

```
[ ]: df_students.groupby('Major')['Math_Score'].mean()
```

```
[ ]: Major
     Computer Science    90.5
     Mathematics         90.0
     Physics             84.5
     Name: Math_Score, dtype: float64
```

```
[ ]: df_students.groupby('Grade_Level').size()
```

```
[ ]: Grade_Level
     High      5
     Medium    1
     dtype: int64
```

1.4 Activity 4: Data Merging

```
[ ]: course_data = {
         'StudentID': [2001, 2002, 2003, 2004, 2005, 2008],
         'Course': ['Python', 'Statistics', 'Physics', 'ML', 'Calculus',␣
      ↪'Chemistry'],
         'Credits': [3, 4, 3, 4, 4, 3]
     }
     df_courses = pd.DataFrame(course_data)
     df_courses
```

```
[ ]:    StudentID      Course  Credits
     0       2001      Python        3
     1       2002  Statistics        4
     2       2003     Physics        3
     3       2004          ML        4
     4       2005    Calculus        4
     5       2008   Chemistry        3
```

```
[ ]: pd.merge(df_students, df_courses, on='StudentID', how='inner')
```

```
[ ]:    StudentID        Name  Age             Major  Math_Score  English_Score  \
     0       2001  Alice Wong   20  Computer Science          85             78
     1       2002    Bob Chen   21       Mathematics          92             85
     2       2003   Carol Lee   19           Physics          78             92
     3       2004   David Kim   22  Computer Science          96             82
     4       2005    Emma Liu   20       Mathematics          88             91

        Total_Score Grade_Level      Course  Credits
     0          163      Medium      Python        3
     1          177        High  Statistics        4
     2          170        High     Physics        3
     3          178        High          ML        4
     4          179        High    Calculus        4
```

```
[ ]: pd.merge(df_students, df_courses, on='StudentID', how='left')
```

```
[ ]:    StudentID        Name  Age             Major  Math_Score  English_Score  \
     0       2001  Alice Wong   20  Computer Science          85             78
     1       2002    Bob Chen   21       Mathematics          92             85
     2       2003   Carol Lee   19           Physics          78             92
     3       2004   David Kim   22  Computer Science          96             82
     4       2005    Emma Liu   20       Mathematics          88             91
     5       2006   Frank Tan   21           Physics          91             86

        Total_Score Grade_Level      Course  Credits
     0          163      Medium      Python      3.0
     1          177        High  Statistics      4.0
```

```
2          170        High     Physics      3.0
3          178        High          ML      4.0
4          179        High    Calculus      4.0
5          177        High         NaN      NaN
```

[ ]:

```
inner_merged = pd.merge(df_students, df_courses, on='StudentID', how='inner')
inner_merged[['StudentID','Name','Course']].drop_duplicates().
 ↪sort_values(by=['StudentID','Course'])
```

[ ]:

```
   StudentID        Name      Course
0       2001  Alice Wong      Python
1       2002    Bob Chen  Statistics
2       2003   Carol Lee     Physics
3       2004   David Kim          ML
4       2005    Emma Liu    Calculus
```

[ ]:

```
inner_merged.groupby(['StudentID','Name'])['Credits'].sum().reset_index().
 ↪rename(columns={'Credits':'Total_Credits'})
```

[ ]:

```
   StudentID        Name  Total_Credits
0       2001  Alice Wong              3
1       2002    Bob Chen              4
2       2003   Carol Lee              3
3       2004   David Kim              4
4       2005    Emma Liu              4
```

[ ]: