pandas-2

February 13, 2025

[1]:
```
# Employee Data Analysis
import pandas as pd

# Creating a sample dataset
data = {
    'Emp_ID': [101, 102, 103, 104, 105],
    'Name': ['Amit', 'Neha', 'Raj', 'Simran', 'Vikas'],
    'Department': ['HR', 'IT', 'Finance', 'IT', 'HR'],
    'Joining_Date': ['2018-06-12', '2020-09-20', '2019-02-15', '2021-07-10',␣
 ↪'2017-11-25'],
    'Salary': [50000, 60000, 55000, 70000, 48000]
}

df = pd.DataFrame(data)

# Convert Joining Date to datetime
df['Joining_Date'] = pd.to_datetime(df['Joining_Date'])

# Filter employees from the IT department
it_employees = df[df['Department'] == 'IT']

# Sort by Joining Date
it_employees = it_employees.sort_values(by='Joining_Date')

# Save the filtered data to a CSV file
it_employees.to_csv("IT_Employees.csv", index=False)

# Display result
print(it_employees)
```

```
   Emp_ID    Name Department Joining_Date  Salary
1     102    Neha         IT   2020-09-20   60000
3     104  Simran         IT   2021-07-10   70000
```

[1]:
```
# Merging Two Student Datasets
import pandas as pd

# Student dataset 1
```

```
data1 = {
    'Student_ID': [1, 2, 3, 4],
    'Name': ['Aarav', 'Pooja', 'Rahul', 'Ananya'],
    'Course': ['B.Tech', 'B.Sc', 'B.Com', 'BBA']
}

# Student dataset 2
data2 = {
    'Student_ID': [3, 4, 5, 6],
    'Name': ['Rahul', 'Ananya', 'Karan', 'Ritika'],
    'Course': ['B.Com', 'BBA', 'BA', 'BCA'],
    'Email': ['rahul@example.com', 'ananya@example.com', 'karan@example.com',␣
 ↪'ritika@example.com']
}

df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)

# Merge datasets based on Student_ID
merged_df = pd.merge(df1, df2, on='Student_ID', how='outer', suffixes=('_old',␣
 ↪'_new'))

# Fill missing values
merged_df.fillna("Not Available", inplace=True)

# Display result
print(merged_df)
```

```
   Student_ID       Name_old     Course_old       Name_new     Course_new  \
0           1          Aarav         B.Tech  Not Available  Not Available
1           2          Pooja           B.Sc  Not Available  Not Available
2           3          Rahul          B.Com          Rahul          B.Com
3           4         Ananya            BBA         Ananya            BBA
4           5  Not Available  Not Available          Karan             BA
5           6  Not Available  Not Available         Ritika            BCA

               Email
0      Not Available
1      Not Available
2   rahul@example.com
3  ananya@example.com
4   karan@example.com
5  ritika@example.com
```

[3]:
```
# Processing Customer Feedback Data
import pandas as pd
```

```
# Creating sample feedback dataset
data = {
    'Customer_ID': [101, 102, 103, 104, 105],
    'Feedback': [
        'Great service and friendly staff!',
        'The product quality is good but expensive.',
        'Delivery was late, not satisfied.',
        'Loved the experience! Will buy again.',
        'Customer support was unresponsive.'
    ]
}

df = pd.DataFrame(data)

# Categorizing feedback as Positive, Neutral, or Negative
def categorize_feedback(text):
    if 'great' in text.lower() or 'loved' in text.lower():
        return 'Positive'
    elif 'late' in text.lower() or 'unresponsive' in text.lower():
        return 'Negative'
    else:
        return 'Neutral'

df['Feedback_Category'] = df['Feedback'].apply(categorize_feedback)

# Save processed data
df.to_csv("Customer_Feedback.csv", index=False)

# Display result
print(df)
```

```
   Customer_ID                                   Feedback Feedback_Category
0          101           Great service and friendly staff!          Positive
1          102  The product quality is good but expensive.           Neutral
2          103           Delivery was late, not satisfied.          Negative
3          104       Loved the experience! Will buy again.          Positive
4          105          Customer support was unresponsive.          Negative
```

[4]:
```
# Movie Dataset Manipulation
import pandas as pd

# Creating sample movie dataset
data = {
    'Movie_ID': [1, 2, 3, 4, 5],
    'Title': ['Movie A', 'Movie B', 'Movie C', 'Movie D', 'Movie E'],
    'Genre': ['Action', 'Drama', 'Comedy', 'Action', 'Drama'],
    'Rating': [4.5, 3.8, 4.2, 4.8, 3.9]
```

```
}

df = pd.DataFrame(data)

# Filtering Action movies
action_movies = df[df['Genre'] == 'Action']

# Sorting by Rating in descending order
action_movies = action_movies.sort_values(by='Rating', ascending=False)

# Save filtered data
action_movies.to_csv("Action_Movies.csv", index=False)

# Display result
print(action_movies)
```

```
   Movie_ID    Title   Genre  Rating
3         4  Movie D  Action     4.8
0         1  Movie A  Action     4.5
```

[5]:
```
# Employee Promotion Analysis
import pandas as pd

# Creating employee dataset
data = {
    'Emp_ID': [101, 102, 103, 104, 105],
    'Name': ['Amit', 'Neha', 'Raj', 'Simran', 'Vikas'],
    'Joining_Year': [2015, 2017, 2018, 2019, 2016],
    'Current_Role': ['Executive', 'Manager', 'Executive', 'Executive',␣
 ↪'Manager']
}

df = pd.DataFrame(data)

# Define promotion criteria (more than 5 years in company)
df['Years_Experience'] = 2024 - df['Joining_Year']
df['Eligible_For_Promotion'] = df['Years_Experience'] > 5

# Save results
df.to_csv("Employee_Promotion.csv", index=False)

# Display result
print(df)
```

```
   Emp_ID    Name  Joining_Year Current_Role  Years_Experience  \
0     101    Amit          2015    Executive                 9
1     102    Neha          2017      Manager                 7
```

```
2     103     Raj          2018    Executive                6
3     104  Simran          2019    Executive                5
4     105   Vikas          2016      Manager                8

   Eligible_For_Promotion
0                    True
1                    True
2                    True
3                   False
4                    True
```

[6]:
```
# Library Book Management System
import pandas as pd

# Creating sample book dataset
data = {
    'Book_ID': [101, 102, 103, 104, 105],
    'Title': ['Python Basics', 'Data Science Guide', 'Machine Learning', 'Web␣
 ↪Development', 'AI Revolution'],
    'Author': ['John Doe', 'Alice Smith', 'Robert Brown', 'Emily Davis',␣
 ↪'Michael Lee'],
    'Stock': [5, 0, 2, 7, 0]
}

df = pd.DataFrame(data)

# Filter books that are available in stock
available_books = df[df['Stock'] > 0]

# Updating stock after issuing a book
df.loc[df['Book_ID'] == 103, 'Stock'] -= 1  # Issuing "Machine Learning" book

# Save the updated dataset
df.to_csv("Library_Books.csv", index=False)

# Display result
print(df)
```

```
   Book_ID               Title        Author  Stock
0      101       Python Basics      John Doe      5
1      102  Data Science Guide   Alice Smith      0
2      103    Machine Learning  Robert Brown      1
3      104     Web Development   Emily Davis      7
4      105       AI Revolution   Michael Lee      0
```

[7]:
```
# Student Attendance Tracker
import pandas as pd
```

```
# Creating sample attendance dataset
data = {
    'Student_ID': [1, 2, 3, 4, 5],
    'Name': ['Aarav', 'Pooja', 'Rahul', 'Ananya', 'Karan'],
    'Total_Classes': [40, 40, 40, 40, 40],
    'Attended_Classes': [38, 30, 25, 35, 20]
}

df = pd.DataFrame(data)

# Calculate attendance percentage
df['Attendance_Percentage'] = (df['Attended_Classes'] / df['Total_Classes']) *␣
 ↪100

# Mark students with low attendance (<75%)
df['Low_Attendance'] = df['Attendance_Percentage'] < 75

# Save result
df.to_csv("Student_Attendance.csv", index=False)

# Display result
print(df)
```

```
   Student_ID    Name  Total_Classes  Attended_Classes  Attendance_Percentage  \
0           1   Aarav             40                38                   95.0
1           2   Pooja             40                30                   75.0
2           3   Rahul             40                25                   62.5
3           4  Ananya             40                35                   87.5
4           5   Karan             40                20                   50.0

   Low_Attendance
0           False
1           False
2            True
3           False
4            True
```

[8]:
```
# E-Commerce Order Processing
import pandas as pd

# Creating sample order dataset
data = {
    'Order_ID': [1001, 1002, 1003, 1004, 1005],
    'Customer_Name': ['Amit', 'Neha', 'Raj', 'Simran', 'Vikas'],
    'Order_Date': ['2024-01-10', '2024-01-15', '2024-01-18', '2024-01-20',␣
 ↪'2024-01-25'],
```

```
    'Delivery_Date': ['2024-01-15', '2024-01-22', None, '2024-01-28', None]
}

df = pd.DataFrame(data)

# Convert dates to datetime format
df['Order_Date'] = pd.to_datetime(df['Order_Date'])
df['Delivery_Date'] = pd.to_datetime(df['Delivery_Date'])

# Calculate delivery time for completed orders
df['Delivery_Time_Days'] = (df['Delivery_Date'] - df['Order_Date']).dt.days

# Identify pending orders
df['Order_Status'] = df['Delivery_Date'].isna()

# Save results
df.to_csv("Ecommerce_Orders.csv", index=False)

# Display result
print(df)
```

```
   Order_ID Customer_Name Order_Date Delivery_Date  Delivery_Time_Days  \
0      1001          Amit 2024-01-10    2024-01-15                 5.0
1      1002          Neha 2024-01-15    2024-01-22                 7.0
2      1003           Raj 2024-01-18           NaT                 NaN
3      1004        Simran 2024-01-20    2024-01-28                 8.0
4      1005         Vikas 2024-01-25           NaT                 NaN

   Order_Status
0         False
1         False
2          True
3         False
4          True
```

[9]:
```
# Social Media User Activity Analysis
import pandas as pd

# Creating sample user activity dataset
data = {
    'User_ID': [101, 102, 103, 104, 105],
    'Username': ['userA', 'userB', 'userC', 'userD', 'userE'],
    'Last_Post_Date': ['2024-02-05', '2024-01-20', '2023-12-10', '2024-02-08',␣
 ↪'2023-11-25'],
    'Total_Posts': [50, 30, 5, 80, 2]
}
```

```
df = pd.DataFrame(data)

# Convert Last_Post_Date to datetime
df['Last_Post_Date'] = pd.to_datetime(df['Last_Post_Date'])

# Find inactive users (last post more than 60 days ago)
df['Inactive'] = (pd.to_datetime("2024-02-10") - df['Last_Post_Date']).dt.days␣
 ↪> 60

# Categorize users based on posts
def categorize_user(posts):
    if posts > 50:
        return 'High Activity'
    elif posts > 20:
        return 'Moderate Activity'
    else:
        return 'Low Activity'

df['Activity_Level'] = df['Total_Posts'].apply(categorize_user)

# Save results
df.to_csv("Social_Media_Users.csv", index=False)

# Display result
print(df)
```

```
   User_ID Username Last_Post_Date  Total_Posts  Inactive     Activity_Level
0      101    userA     2024-02-05           50     False  Moderate Activity
1      102    userB     2024-01-20           30     False  Moderate Activity
2      103    userC     2023-12-10            5      True       Low Activity
3      104    userD     2024-02-08           80     False      High Activity
4      105    userE     2023-11-25            2      True       Low Activity
```

[10]:
```
# Customer Support Ticket Analysis
import pandas as pd

# Creating sample ticket dataset
data = {
    'Ticket_ID': [501, 502, 503, 504, 505],
    'Customer_Name': ['Amit', 'Neha', 'Raj', 'Simran', 'Vikas'],
    'Issue_Type': ['Billing', 'Technical', 'Account', 'Technical', 'Billing'],
    'Resolved': [True, False, True, False, True],
    'Priority': [None, None, None, None, None]
}

df = pd.DataFrame(data)
```

```python
# Assign priority based on issue type
def assign_priority(issue):
    if issue == 'Technical':
        return 'High'
    elif issue == 'Billing':
        return 'Medium'
    else:
        return 'Low'

df['Priority'] = df['Issue_Type'].apply(assign_priority)

# Identify unresolved tickets
df['Unresolved'] = ~df['Resolved']

# Save results
df.to_csv("Support_Tickets.csv", index=False)

# Display result
print(df)
```

```
   Ticket_ID Customer_Name Issue_Type  Resolved Priority  Unresolved
0        501          Amit    Billing      True   Medium       False
1        502          Neha  Technical     False     High        True
2        503           Raj    Account      True      Low       False
3        504        Simran  Technical     False     High        True
4        505         Vikas    Billing      True   Medium       False
```

[ ]: