
Pandas Cheatsheet

Scientific Programming

1. Import Pandas

```
import pandas as pd
```

2. Creating a Series

```
s = pd.Series([10, 20, 30, 40],
              index=['a', 'b', 'c', 'd'])
```

	0
a	10
b	20
c	30
d	40

3. Creating a DataFrame

```
df = pd.DataFrame({'Age': [28, 24, 35, 28],
                  'City': ['York', 'Oslo', 'Lima', 'York']},
                  index = ['John', 'An', 'Peter', 'Tom'])
```

	Age	City
John	28	York
An	24	Oslo
Peter	35	Lima
Tom	28	York

4. Column Operations: Adding, Multiplying

```
# Adding a new column
```

```
df['Salary'] = [50000, 60000, 75000, 55000]
```

```
# Compute new column
```

```
df['Bonus'] = df['Salary'] * 0.10
```

```
# Adding, subtracting, multiplying columns
```

```
df['Total'] = df['Salary'] + df['Bonus']
```

```
df['Years Until Retirement'] = 65 - df['Age']
```

5. Reading and Writing

```
df = pd.read_csv('data.csv')
df.to_csv('output.csv', index=False)
```

6. Accessing Data (Indexing and Slicing)

```
# Accessing rows:
```

```
df.iloc[3] # Access by position
```

```
df.loc['Peter'] # Access by index label
```

```
# Multiple indices
```

```
df.iloc[[2, 3]]
```

```
df.loc[['An', 'Peter']]
```

```
# Slicing rows
```

```
df.iloc[0:3:2] # Rows 0 to 3 (steps of 2)
```

```
# Accessing columns
```

```
df['Age'] # Single column -> Series
```

```
df[['City', 'Age']] # Multiple columns -> DF
```

7. Statistics

```
df['Age'].mean() df['Age'].min()
```

```
df['Age'].median() df['Age'].max()
```

```
df['Age'].sum() df['Age'].std()
```

```
df['Age'].count() # Count non-null values
```

8. Filtering Data

```
# Filter rows based on a single condition.
```

```
df[df['Age'] > 30]
```

```
# Combine multiple conditions using & and |
```

```
filtered_df = df[
    (df['Age'] > 30) & (df['City'] == 'Lima')]
```

```
filtered_df = df[
    (df['Age'] < 30) | (df['City'] == 'York')]
```

9. Grouping

```
grouped = df.groupby('City')
```

```
# Loop over groups
```

```
for city, group in grouped:
    print(f"\nCity: {city}")
    print(group)
```

10. Aggregation Functions

```
grouped['Age'].mean() grouped['Age'].min()
grouped['Age'].median() grouped['Age'].max()
grouped['Age'].sum() grouped['Age'].std()
```

```
grouped['Age'].count() # Count non-null values
grouped.head(n) # First n rows per group
```

11. Sorting Data

```
# Sort by a single column
```

```
df.sort_values(by='Age')
```

```
# Sort by multiple columns
```

```
df.sort_values(by=['Age', 'Salary'],
               ascending=[True, False])
```

12. Combining DataFrames

```
# Combine horizontally
```

```
combined = pd.merge(df1, df2, on='City',
                    how='left') # 'inner'/'outer'/'left'/'right'
```

```
# Combine vertically
```

```
combined = pd.concat([df1, df2])
```
