

# Pandas - Data Selection and Filtering

## Basic Column Selection

```
import pandas as pd
import numpy as np

df = pd.DataFrame({
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [25, 30, 35, 28],
    'Score': [85, 92, 78, 88],
    'City': ['NYC', 'LA', 'Chicago', 'Boston']
})

# Single column
print("Ages:", df['Age'].values)

# Multiple columns
print("\nName and Score:")
print(df[['Name', 'Score']])

# All columns except one
print("\nAll except City:")
print(df.drop('City', axis=1))
```

## Row Filtering

```
# Filter by condition
high_scorers = df[df['Score'] > 80]
print("High scorers (>80):")
print(high_scorers)

# Multiple conditions with &
young_high = df[(df['Age'] < 30) & (df['Score'] > 80)]
print("\nYoung high scorers:")
print(young_high)

# Multiple conditions with |
filter_or = df[(df['Age'] < 27) | (df['Score'] > 90)]
print("\nAge < 27 OR Score > 90:")
print(filter_or)

# Using isin()
cities = ['NYC', 'LA']
city_filter = df[df['City'].isin(cities)]
print("\nFrom NYC or LA:")
print(city_filter)
```

## loc and iloc Indexing

```
# loc: label-based indexing
print("Using loc[0] (by label):")
print(df.loc[0])

print("\nloc[0:2, 'Name':'Age']:")
print(df.loc[0:2, 'Name':'Age'])

# iloc: integer-based indexing
print("\nUsing iloc[0] (by position):")
print(df.iloc[0])
```

```

print("\niloc[0:2, 0:2]:")
print(df.iloc[0:2, 0:2])

# Combination
print("\nFirst 2 rows, specific columns:")
print(df.loc[:1, ['Name', 'Score']])

```

## Advanced Filtering Techniques

```

# String operations
print("Names starting with 'A' or 'B':")
print(df[df['Name'].str.startswith(('A', 'B'))])

# Between values
print("\nScores between 80 and 90:")
print(df[df['Score'].between(80, 90)])

# Query method
print("\nUsing query:")
print(df.query('Age > 25 and Score > 80'))

# Negation
print("\nNot from NYC:")
print(df[~df['City'].str.contains('NYC')])

```

## Complex Selection Patterns

```

# Select rows and modify
df_copy = df.copy()
df_copy.loc[df_copy['Score'] > 85, 'Grade'] = 'A'
df_copy.loc[df_copy['Score'] <= 85, 'Grade'] = 'B'
print("With grades:")
print(df_copy)

# Select based on index
print("\nSelect specific indices:")
print(df.iloc[[0, 2]])

# Conditional selection with multiple columns
mask = (df['Age'] > 25) & (df['Score'] > 80)
print("\nCustom mask:")
print(df.loc[mask, ['Name', 'Age', 'Score']])

```