



# **Linux Command Line & UV**

Cơ Bản Cho Data Science/AI

---

Quanskill Academy

Giới Thiệu Linux Command Line

Các Lệnh Cơ Bản

Làm Việc Với Files

Package Management với UV

Workflow Data Science

Tips & Tricks

Thực Hành

Kết Luận

# **Giới Thiệu Linux Command Line**

---

Terminal (hay Command Line Interface - CLI) là cách tương tác với máy tính bằng text thay vì giao diện đồ họa.

## Tại sao Data Scientists cần biết Linux?

- Hầu hết servers chạy Linux
- Xử lý dữ liệu lớn hiệu quả hơn
- Automation và scripting
- Quản lý environments và packages

## Lưu Ý

Mọi lệnh sẽ được giải thích từng bước với output cụ thể!

## Cách mở Terminal:

- **Linux/Mac:** Ctrl+Alt+T hoặc tìm "Terminal"
- **Windows:** WSL, Git Bash, hoặc PowerShell

Khi mở terminal, bạn sẽ thấy:

```
1 username@hostname: ~ $
```

Giải thích:

- **username:** tên người dùng
- **hostname:** tên máy tính
- **~:** thư mục home
- **\$:** prompt (sẵn sàng nhận lệnh)

## Các Lệnh Cơ Bản

---

## Lệnh pwd (Print Working Directory)

```
1 $ pwd  
2 /home/username
```

**Output:** Hiển thị đường dẫn thư mục hiện tại

### Ví dụ thực tế:

```
1 $ pwd  
2 /home/datascientist/projects/ml-project
```

### Giải thích

Bạn đang ở trong thư mục ml-project bên trong projects của user datascientist

## Lệnh ls (List)

```
1 $ ls
2 data  models  notebooks  requirements.txt
```

## Các tùy chọn hữu ích:

```
1 $ ls -l    # Long format - chi tiết
2 -rw-r--r-- 1 user user 1024 Dec 15 10:30 requirements.txt
3 drwxr-xr-x 2 user user 4096 Dec 15 09:15 data/
4
5 $ ls -la   # Hiện thi file an
6 .env  .gitignore  data/  models/  notebooks/
7
8 $ ls -lh   # Human readable file sizes
9 -rw-r--r-- 1 user user 1.5K Dec 15 10:30 requirements.txt
10 drwxr-xr-x 2 user user 4.0K Dec 15 09:15 data/
```

## Lệnh cd (Change Directory)

```
1 $ pwd  
2 /home/datascientist  
3  
4 $ cd projects  
5 $ pwd  
6 /home/datascientist/projects  
7  
8 $ cd ml-project  
9 $ pwd  
10 /home/datascientist/projects/ml-project  
11  
12 $ cd ..      # Go back one level  
13 $ pwd  
14 /home/datascientist/projects  
15  
16 $ cd ~      # Go to home directory  
17 $ pwd  
18 /home/datascientist
```

## Ký hiệu đặc biệt

.. = thư mục cha, . = thư mục hiện tại, ~ = thư mục home

## Lệnh mkdir (Make Directory)

```
1 $ mkdir my-data-project
2 $ ls
3 my-data-project/
4
5 $ mkdir -p deep/nested/folders
6 $ ls deep/
7 nested/
8
9 $ ls deep/nested/
10 folders/
```

## Ví dụ tạo cấu trúc project:

```
1 $ mkdir -p ml-project/{data,notebooks,models,src}
2 $ ls ml-project/
3 data/  models/  notebooks/  src/
```

## Giải thích

-p tạo các thư mục cha nếu chưa tồn tại

## Làm Việc Với Files

---

## Lệnh touch tạo file rỗng

```
1 $ touch main.py
2 $ ls -l main.py
3 -rw-r--r-- 1 user user 0 Dec 15 11:00 main.py
4
5 $ touch data.csv model.pkl requirements.txt
6 $ ls
7 data.csv  main.py  model.pkl  requirements.txt
```

## Tạo nhiều file cho Data Science project:

```
1 $ touch {train,test,validation}.csv
2 $ ls *.csv
3 test.csv  train.csv  validation.csv
4
5 $ touch {EDA,model_training,evaluation}.ipynb
6 $ ls *.ipynb
7 EDA.ipynb  evaluation.ipynb  model_training.ipynb
```

## Lệnh cat hiển thị nội dung file

Tạo file mẫu:

```
1 $ echo "pandas==1.5.0" > requirements.txt
2 $ echo "numpy==1.24.0" >> requirements.txt
3 $ echo "scikit-learn==1.2.0" >> requirements.txt
```

Đọc nội dung:

```
1 $ cat requirements.txt
2 pandas==1.5.0
3 numpy==1.24.0
4 scikit-learn==1.2.0
```

## Giải thích

> ghi đè file, » thêm vào cuối file

Rất hữu ích khi làm việc với dataset lớn!

Tạo file CSV mẫu:

```
1 $ echo "name,age,salary" > employees.csv
2 $ echo "Alice,25,50000" >> employees.csv
3 $ echo "Bob,30,60000" >> employees.csv
4 $ echo "Charlie,35,70000" >> employees.csv
5 $ echo "Diana,28,55000" >> employees.csv
6 $ echo "Eve,32,65000" >> employees.csv
```

Xem 3 dòng đầu:

```
1 $ head -3 employees.csv
2 name,age,salary
3 Alice,25,50000
4 Bob,30,60000
```

Xem 2 dòng cuối:

```
1 $ tail -2 employees.csv
2 Diana,28,55000
3 Eve,32,65000
```

## Package Management với UV

---

**UV** là package manager hiện đại cho Python, nhanh hơn pip rất nhiều!

## Ưu điểm:

- Cài đặt packages nhanh hơn pip 10-100 lần
- Quản lý virtual environments tốt hơn
- Lock files tự động
- Compatible với pip/PyPI

## Cài đặt UV:

```
1 # On Linux/Mac
2 $ curl -LsSf https://astral.sh/uv/install.sh | sh
3
4 # Verify installation
5 $ uv --version
6 uv 0.1.0
```

## Bước 1: Tạo project mới

```
1 $ uv init ml-analysis-project
2 $ cd ml-analysis-project
3 $ ls
4 README.md pyproject.toml src/
```

## Bước 2: Xem cấu trúc project

```
1 $ cat pyproject.toml
2 [project]
3 name = "ml-analysis-project"
4 version = "0.1.0"
5 description = ""
6 dependencies = []
7
8 [build-system]
9 requires = ["hatchling"]
10 build-backend = "hatchling.build"
```

## Giải thích

UV tự động tạo cấu trúc project chuẩn với pyproject.toml

## Cài đặt các packages cơ bản:

```
1 $ uv add pandas numpy matplotlib seaborn
2 Resolved 8 packages in 145ms
3   Built ml-analysis-project @ file:///path/to/project
4 Prepared 4 packages in 1.23s
5 Installed 8 packages in 45ms
6   + matplotlib==3.8.2
7   + numpy==1.26.2
8   + pandas==2.1.4
9   + seaborn==0.13.0
10  + ...
```

## Xem dependencies đã cài:

```
1 $ uv tree
2 ml-analysis-project v0.1.0
3   +-- matplotlib v3.8.2
4   +-- numpy v1.26.2
5   +-- pandas v2.1.4
6   +-- seaborn v0.13.0
```

## So sánh

Cùng một việc với pip có thể mất 30-60 giây, UV chỉ mất vài giây!

## UV tự động quản lý virtual environment!

Chạy Python với environment của project:

```
1 $ uv run python
2 Python 3.12.0 (main, Oct 2 2023, 13:45:54)
3 >>> import pandas as pd
4 >>> import numpy as np
5 >>> print("Data Science packages loaded successfully!")
6 Data Science packages loaded successfully!
7 >>> exit()
```

Chạy script Python:

```
1 $ echo 'import pandas as pd
2 print("Pandas version:", pd.__version__)
3 df = pd.DataFrame({"A": [1,2,3], "B": [4,5,6]})
4 print(df)' > analysis.py
5
6 $ uv run python analysis.py
7 Pandas version: 2.1.4
8      A   B
9      0   1   4
10     1   2   5
11     2   3   6
```

# Workflow Data Science

---

## Bước 1: Tạo cấu trúc thư mục

```
1 $ mkdir -p data-science-project/{data/raw,data/processed,notebooks,src,models,reports}
2 $ cd data-science-project
3 $ tree
4 .
5     data/
6         processed/
7             raw/
8         models/
9         notebooks/
10        reports/
11        src/
```

## Bước 2: Khởi tạo UV project

```
1 $ uv init .
2 $ uv add pandas numpy matplotlib seaborn jupyter scikit-learn
```

## Tạo file CSV mẫu để thực hành:

```
1 $ cat > data/raw/sales_data.csv << EOF
2 date,product,quantity,price,revenue
3 2023-01-01,Laptop,5,1000,5000
4 2023-01-02,Mouse,20,25,500
5 2023-01-03,Keyboard,15,50,750
6 2023-01-04,Monitor,3,300,900
7 2023-01-05,Laptop,2,1000,2000
8 EOF
9
10 $ head data/raw/sales_data.csv
11 date,product,quantity,price,revenue
12 2023-01-01,Laptop,5,1000,5000
13 2023-01-02,Mouse,20,25,500
```

## Kiểm tra kích thước file:

```
1 $ wc -l data/raw/sales_data.csv
2 6 data/raw/sales_data.csv # 5 data + 1 header
```

## Khởi động Jupyter với UV:

```
1 $ uv run jupyter notebook
2 [I 2023-12-15 10:30:00.000 ServerApp] Serving at http://localhost:8888
3 [I 2023-12-15 10:30:00.000 ServerApp] Use Control-C to stop this server
```

## Hoặc tạo notebook từ command line:

```
1 $ touch notebooks/data_exploration.ipynb
2 $ uv run jupyter notebook notebooks/data_exploration.ipynb
```

## Lưu Ý

Jupyter sẽ mở browser tự động. Tất cả packages đã cài với UV đều available trong notebook!

## Tips & Tricks

---

## Keyboard shortcuts trong Terminal:

- Ctrl+C: Dừng lệnh đang chạy
- Ctrl+L: Clear screen (hoặc clear)
- Tab: Auto-complete tên file/folder
- ↑/↓: Lịch sử lệnh đã chạy
- Ctrl+R: Search lịch sử lệnh

## Ví dụ auto-complete:

```
1 $ cd data # Type "cd da" then press Tab
2 $ ls sales_ # Type "ls sa" then press Tab
3 sales_data.csv
```

## Ví dụ tìm lệnh cũ:

```
1 $ # Press Ctrl+R then type "uv add"
2 (reverse-i-search)‘uv add’: uv add pandas numpy matplotlib
```

## Wildcard patterns:

```
1 $ ls *.csv          # All CSV files
2 train.csv  test.csv  validation.csv
3
4 $ ls data/*.py      # All Python files in data folder
5 data/clean.py  data/preprocess.py
6
7 $ ls models/model_* # Files starting with "model_"
8 models/model_v1.pkl  models/model_v2.pkl
```

## Pipe và redirection:

```
1 $ ls -la | grep ".csv"      # Find CSV files in detailed list
2 -rw-r--r-- 1 user user 1234 Dec 15 10:30 train.csv
3
4 $ wc -l *.py > line_counts.txt # Count lines in Python files
5 $ cat line_counts.txt
6   50 main.py
7   120 utils.py
8   200 model.py
9   370 total
```

## Quản lý dependencies:

```
1 # Add dev dependencies
2 $ uv add --dev pytest black flake8
3 $ uv add --dev jupyterlab
4
5 # Remove package
6 $ uv remove seaborn
7
8 # Update all packages
9 $ uv lock --upgrade
10
11 # Export to requirements.txt (compatibility)
12 $ uv export --format requirements-txt > requirements.txt
```

## Multiple Python versions:

```
1 # Use specific Python version
2 $ uv python install 3.11
3 $ uv init --python 3.11 old-python-project
4
5 # List available Python versions
6 $ uv python list
```

## **Thực Hành**

---

## Tạo một Data Science project hoàn chỉnh:

1. Tạo thư mục customer-analysis
2. Setup UV project với packages: pandas, numpy, matplotlib, seaborn, scikit-learn
3. Tạo cấu trúc thư mục chuẩn
4. Tạo file CSV với customer data (10 rows)
5. Viết Python script đọc và hiển thị basic statistics
6. Chạy script bằng UV

## Challenge

Hoàn thành trong 10 phút bằng command line only!

```
1 # Step 1-2: Create and setup project
2 $ mkdir customer-analysis && cd customer-analysis
3 $ uv init .
4 $ uv add pandas numpy matplotlib seaborn scikit-learn
5
6 # Step 3: Create folder structure
7 $ mkdir -p {data/raw,data/processed,notebooks,src,models,reports}
8
9 # Step 4: Create sample data
10 $ cat > data/raw/customers.csv << EOF
11 customer_id,name,age,income,spending_score
12 1,Alice,25,50000,70
13 2,Bob,30,60000,80
14 3,Charlie,35,70000,60
15 4,Diana,28,55000,75
16 5,Eve,32,65000,85
17 6,Frank,45,80000,50
18 7,Grace,38,75000,65
19 8,Henry,29,58000,78
20 9,Ivy,33,68000,82
21 10,Jack,41,72000,55
22 EOF
```

# Solution - Phần 2

```
1 # Step 5: Create analysis script
2 $ cat > src/analyze.py << 'EOF'
3 import pandas as pd
4 import numpy as np
5
6 # Load data
7 df = pd.read_csv('data/raw/customers.csv')
8
9 print("==== Customer Analysis Report ===")
10 print(f"Total customers: {len(df)}")
11 print(f"Average age: {df['age'].mean():.1f}")
12 print(f"Average income: ${df['income'].mean():,.0f}")
13 print(f"Average spending score: {df['spending_score'].mean():.1f}")
14
15 print("\n==== Age Distribution ===")
16 print(df['age'].describe())
17
18 print("\n==== Top 5 Customers by Spending Score ===")
19 print(df.nlargest(5, 'spending_score')[['name', 'spending_score']])
20 EOF
```

```
1 # Step 6: Run with UV
2 $ uv run python src/analyze.py
```

## Kết Luận

---

## Những gì chúng ta đã học:

- Linux command line cơ bản: pwd, ls, cd, mkdir
- Làm việc với files: touch, cat, head, tail
- UV package manager: nhanh, hiệu quả, modern
- Setup Data Science project hoàn chỉnh
- Workflow từ setup đến analysis

## Next Steps:

- Thực hành thêm với datasets thực tế
- Tìm hiểu Git cho version control
- Docker cho containerization
- Cloud deployment

Câu Hỏi?

```
1 # Navigation
2 pwd                      # Show current directory
3 ls                        # List files
4 ls -la                    # List with details and hidden files
5 cd folder_name            # Change directory
6 cd ..                     # Go to parent directory
7 cd ~                      # Go to home directory
8
9 # File operations
10 touch filename            # Create empty file
11 mkdir folder_name          # Create directory
12 mkdir -p path/to/dir      # Create nested directories
13 cat filename              # Show file content
14 head -n filename          # Show first n lines
15 tail -n filename          # Show last n lines
16
17 # Useful
18 clear                     # Clear screen
19 history                  # Show command history
```

```
1 # Project setup
2 uv init project_name          # Create new project
3 uv init .                      # Initialize in current directory
4
5 # Package management
6 uv add package_name            # Add package
7 uv add --dev package_name      # Add dev dependency
8 uv remove package_name         # Remove package
9 uv tree                         # Show dependency tree
10
11 # Running
12 uv run python script.py       # Run Python script
13 uv run jupyter notebook        # Start Jupyter
14 uv run python -m module        # Run module
15
16 # Environment
17 uv export > requirements.txt  # Export dependencies
18 uv sync                          # Sync dependencies
19 uv lock --upgrade               # Update lock file
```