Essential Unix Command Tutorial (with Dummy Data & Piping)

A concise, hands-on guide to core Unix commands — ideal for beginners in bioinformatics, data science, or anyone learning the command line.

---

1. Setup and Dummy Dataset

Create a Practice Folder

```
mkdir unix_practice
cd unix_practice
```

- mkdir → makes a new directory.
- cd → changes into that directory.

Create a Dummy Data File

```
cat > dummy_data.txt <<EOF
ID  Name    Department  Salary  Location
101 Alice   Biology 52000   New_York
102 Bob Chemistry   49000   Boston
103 Charlie Physics 51000   Chicago
104 Diana   Mathematics 55000   Seattle
105 Eric    Biology 58000   San_Francisco
106 Fiona   Chemistry   61000   New_York
107 George  Physics 48000   Boston
108 Hannah  Mathematics 53000   Chicago
EOF
```

Creates a simple dataset to practice with.

---

2. File and Directory Management

List Files

```
ls -l
```

Displays files in the current directory with details like permissions, owner, and size.

Copy Files

```
cp dummy_data.txt data_copy.txt
```

Copies one file to another.

Move or Rename Files

```
mv data_copy.txt employees.txt
```

Moves or renames a file.

Delete Files/Folders

```
rm employees.txt
rm -r old_folder/
```

Deletes a file or a folder permanently.

Print Current Directory

```
pwd
```

Shows the full path of your current directory.

Create Empty File

```
touch notes.txt
```

Creates an empty file or updates its timestamp.

3. Viewing and Inspecting Files

View Top or Bottom Lines

```
head -5 dummy_data.txt
tail -3 dummy_data.txt
```

- head → shows the first few lines.
- tail → shows the last few lines.

Count Lines

```
wc -l dummy_data.txt
```

Counts the total number of lines in the file.

Search for Text

```
grep "Biology" dummy_data.txt
```

Finds lines containing “Biology”.

Useful Options:

- -i → ignore case
- -v → exclude lines with pattern
- -n → show line numbers

Example:

```
grep -v "Physics" dummy_data.txt
```

→ shows all lines except those with “Physics”.

---

4. Extracting and Sorting Data

Extract Specific Columns

```
cut -f1,2 dummy_data.txt
```

Displays only columns 1 and 2 (ID and Name). Use -d to change the delimiter.

Sort Data

```
sort -t$'\t' -k4,4n dummy_data.txt
```

Sorts the file numerically by the 4th column (Salary).

Options:

- -t → specify delimiter
- -k → choose column(s)
- -n → numeric sort
- -r → reverse order

---

5. Writing and Redirecting Output

Echo Text

```
echo "Hello Unix world!"
```

Prints text to the terminal.

Redirect Output

```
echo "todo: analyze salaries" > notes.txt
echo "todo: filter by department" >> notes.txt
```

- **>** → overwrite file
- **>>** → append to file

6. Combining Commands with Pipes

Using the Pipe Operator (|)

The pipe (|) connects two commands — sending the output of one as the input of another.

Examples:

```
grep "Biology" dummy_data.txt | wc -l
```

→ Counts how many lines contain “Biology”.

```
cut -f3 dummy_data.txt | sort | uniq
```

→ Extracts the Department column, sorts it, and removes duplicates.

```
grep "New_York" dummy_data.txt | sort -t$'\t' -k4,4nr
```

→ Filters rows for New York employees and sorts by Salary (descending).

Tip: You can chain multiple pipes for powerful one-line data processing.

---

7. Summary of Commands

Command	Description
`mkdir`	Create new directories
`cd`	Change current directory
`pwd`	Show current path
`ls`	List files in a directory
`cp`	Copy files
`mv`	Move or rename files
`rm`	Delete files
`touch`	Create empty files
`head`	Display first lines of a file
`tail`	Display last lines of a file
`wc -l`	Count number of lines
`grep`	Search for text patterns
`cut`	Extract specific columns
`sort`	Sort data by fields
`echo`	Print or write text
`|` (pipe)	Combine multiple commands by passing output to input