

# Unix Utilities in Data Management

Statistical Consulting Group  
IDRE/ATS UCLA

February 4, 2009

# Introduction

- ▶ Huge ASCII data sets - fixed or free format
- ▶ Cluster computing - CCPR or Hoffman2
- ▶ PC/Mac users
- ▶ Getting to know your environment
- ▶ Doing some basic data management tasks more efficiently

- ▶ Making a directory: **mkdir**
  - ▶ **mkdir dms**
- ▶ Changing to a directory: **ch dir**
  - ▶ **cd dms** – change to dms directory
  - ▶ **cd ..** – change to one-level up
  - ▶ **cd /u/local/bin**
  - ▶ **cd** – change back to home directory
  - ▶ **cd ~/dms** – change to dms directory from anywhere
- ▶ Print working directory: **pwd**
- ▶ Remove a directory: **rmdir**

# Directories and Files - continued

- ▶ List files in a directory: **ls**
  - ▶ **ls** – list files in a directory
  - ▶ **ls -l** – list files in a directory and show file attributes
  - ▶ **ls \*.txt** – list files with .txt file extension
  - ▶ **ls -a** – list ALL files in a directory
- ▶ Move file/files to another location
  - ▶ **mv \*.txt ../dms** –move files with .txt extension in the current directory to a dms folder one level up
- ▶ Remove files: **rm \*.dat**
- ▶ Find files: **find**
  - ▶ **find . -name "\*.txt"** –list names of all the files in the current directory and subdirectories with .txt file extension

- ▶ See the content of a file one screen at a time: **more** or **less**
  - ▶ **more logfile.txt** use space bar to view next page and **q** to quit
  - ▶ **less logfile.txt** use space bar to view next page, **ctrl-b** to go back and **q** to quit
- ▶ Output the first part of a file: **head**
  - ▶ **head data1.txt** – display the first 10 lines
  - ▶ **head -5 data1.txt** – display the first 5 lines
  - ▶ **head -5 data1.txt > data1\_5a.txt** – create a file containing the first 5 lines
- ▶ Output the last part of a file: **tail**
  - ▶ **tail data1.txt** – display the last 10 lines
  - ▶ **tail -5 data1.txt** – display the last 5 lines
  - ▶ **tail -5 data1.txt > data1\_5o.txt** –create a file containing the last 5 lines

# Command wc for counting

- ▶ Count the number of rows in a file: **wc -l data1.txt**
- ▶ Count the number of characters in the longest line: **wc -L data1.txt**
- ▶ Count the number of columns in a file with fixed number of columns: **head -1 data1.txt | wc -w**

# Merging and stacking files

- ▶ Merge lines of files: **paste S\*.txt > all\_cols.txt**
- ▶ Merge lines of files in csv format: **paste -d",," S\*.txt > all\_cols.txt**
- ▶ Merge lines in serial (transposed) **paste -s S\*.txt > all\_rows.txt**
- ▶ Join the common lines of sorted two files:  
**sort score.txt > score\_s.txt**  
**sort grade.txt > grade\_s.txt**  
**join score\_s.txt grade\_s.txt > final.txt**
- ▶ Stack files **cat S\*.txt > all\_single.txt**

- ▶ Split a file into smaller files: **split largefile.dat sf**
  - ▶ The default is 1000 lines per file.
  - ▶ Multiple files will be created with prefix **sf**.
  - ▶ Every smaller file will have exactly 1000 lines, except possibly the last one.
- ▶ Select a list of columns or fields from a data file: **cut**
  - ▶ **cut -d" " -f2 grade.txt >grade\_2.txt** –extracting the second field
  - ▶ **cut -c2-3 fixed.dat >fixed\_c2\_c3.txt** –extracting the second and the third column

## Splitting files - continued

Let's say a file has  $N$  lines and we want to create a new file starting from the  $n$ th line and ending with  $m$ th line with  $m > n$ .

- ▶ Will use **tail** and **head** command for this task.
- ▶ For **tail** going backward, the parameter is  $N - n$ .
- ▶ For **head** going forward in the subset, the parameter is  $m - n + 1$ .
- ▶ **tail**  $-(N - n)$  **largedata.dat** > **trimmed.txt**
- ▶ **head**  $-(m - n + 1)$  **trimmed.txt** > **final.txt**

# Searching for a string

- ▶ Search one or more files for lines that match a literal, text-string pattern: **fgrep**
  - ▶ **fgrep "mark" \*.txt** – search all .txt files for lines containing the string *mark*
- ▶ Search one or more files for lines that match a regular expression regexp: **grep**
  - ▶ **grep "[a-zA-Z?]" problem.txt > pline.txt** –search all lines containing characters.

# Using command -for-

The command **for** is very handy for doing repeated tasks.

- ▶ **for FILE in \*.txt do wc -l \$FILE; done;** - counting number of lines for all the .txt files

# Creating and running a batch file

```
#!/bin/bash

seq 10 > small10.txt
more small10.txt
date
ls -l S*.txt
```

- ▶ Save the content to a file with .bat extension, say myjob.bat
- ▶ Change mode to make **myjob.bat** executable (**chmod 777 myjob.bat**)
- ▶ Run it using **./myjob.bat**

# When things go wrong...

- ▶ **ctrl-c** to stop a job
- ▶ **ctrl-z** to suspend a job
- ▶ **ps** to list the running jobs
- ▶ **kill -9** to kill a job

# Unix-like commands for Windows

- ▶ Almost everything discussed here works under Windows environment with GNU utilities.
- ▶ <http://sourceforge.net/projects/unxutils>
- ▶ [http://www.ats.ucla.edu/stat/fileman/unix\\_cmds.htm](http://www.ats.ucla.edu/stat/fileman/unix_cmds.htm)
- ▶ [http://www.ats.ucla.edu/stat/fileman/what\\_canbe\\_done.htm](http://www.ats.ucla.edu/stat/fileman/what_canbe_done.htm)