Working with Data in the Shell

Spring 2016 BZAN 583

Drew Schmidt

February 01, 2016

Today's Learning Objectives

- Learn about variables in the shell.
- See some techniques for basic data analysis.
- See supplemental document if you want to know more about shell programming.

Quick Note on Scripting

Scripting in a Slide

- Just the very basics (see handout for details).
- Place your commands in a file.
- Execute `chmod +x myfile`.
- Run the script via:
    - relative path: `./myfile`
    - absolute path: `/path/to/myfile`
- The first line can be a "magic comment"
    - `#!/bin/sh`
    - `#!/bin/sh`
- Conditionals and loops are possible, but beyond scope.

Variables

- Assign with =
- If a space occurs, put the RHS in quotes
- Reference with preceeding $
- Store the output of a command with “backticks” ‘

Example

```
x=something
x = something # NO!

x="something else"

echo $x

x=`ls ~ | grep D`
echo $x
```

Getting Data and Inspecting

Data Download/Transfer Tools

- `wget`
- `curl`
- `sftp`

Extracting Data from Archives

- `tar zxf archive.tar.gz`
- `gunzip archive.gz`
- `unzip archive.zip`

Basic Inspection Tools

- `head`/`tail`
- `grep`
- `less`

Example

```
head -2 pop.csv founded.csv

grep -i nashville *.csv

less diamonds.csv
```

Processing Data

Some Advice

- Downsampling? Try `grep`.
- Editing entries? Try `sed`.
- Working with columns? `awk`, plus some others.
- The more complicated your task, the less suitable the shell is for it!
- For simple tasks, very powerful.

Dropping Lines

```
sed -i /^$/d diamonds.csv | head

sort -u diamonds.csv | head

tail -n +2 diamonds.csv | head
```

Convert CSV to TSV

```
sed 's/,/\t/g' diamonds.csv | head
sed 's/,/\t/g' diamonds.csv > diamonds.tsv

awk 'BEGIN {FS=","; OFS="\t"} {$1=$1; print}' diamonds.csv | h
```

A Word of Caution

- I claim “there is no such thing as a CSV”.
- *Regular expressions are not substitutes for parsers.*
- Consider: `1,"2,\"3,4\",5",6`
  - How many fields would most people say?
  - How many fields does `sed` say?
- Nothing is ever easy. Think about what you’re doing!

Dropping a Variable

```
cut -f 3 diamonds.tsv  | head
cut --complement -f 3 diamonds.tsv  | head

cut --complement -f 3 -d, diamonds.csv  | head

mv diamonds.csv diamonds.csv.old
cut --complement -f 3 -d, diamonds.csv.old > diamonds.csv
rm diamonds.csv.old
```

Subsetting Diamonds

```
grep Premium diamonds.tsv | head

grep -v Premium diamonds.tsv | head

grep "Premium\|Very Good" diamonds.tsv | head
```

Combining Files

Combining Files

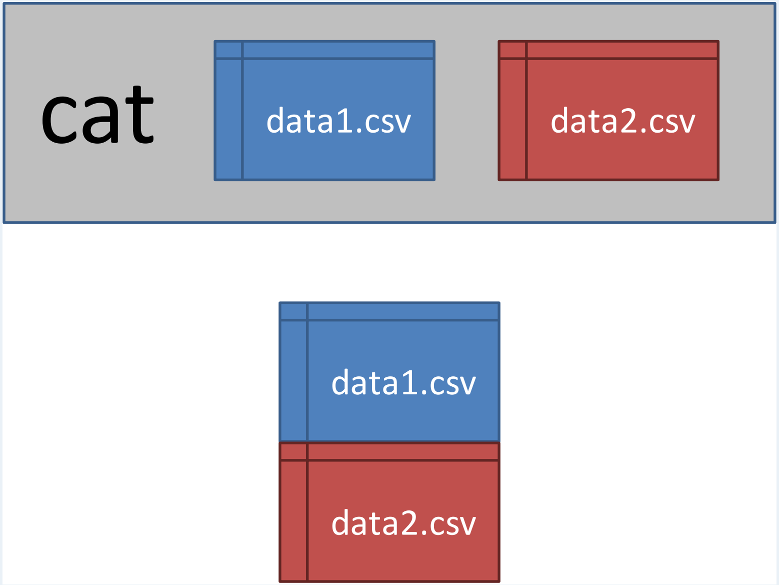
- `cat` and `crop`
- Handles simple things very well.
- For complex tasks, use the appropriate tool.

Example Data

```
cat pop.csv
## City,Metro Population
## Knoxville,852715
## Nashville,1757912
## Memphis,1341746

cat founded.csv
## City,Founded
## Knoxville,1791
## Nashville,1779
## Memphis,1819
```

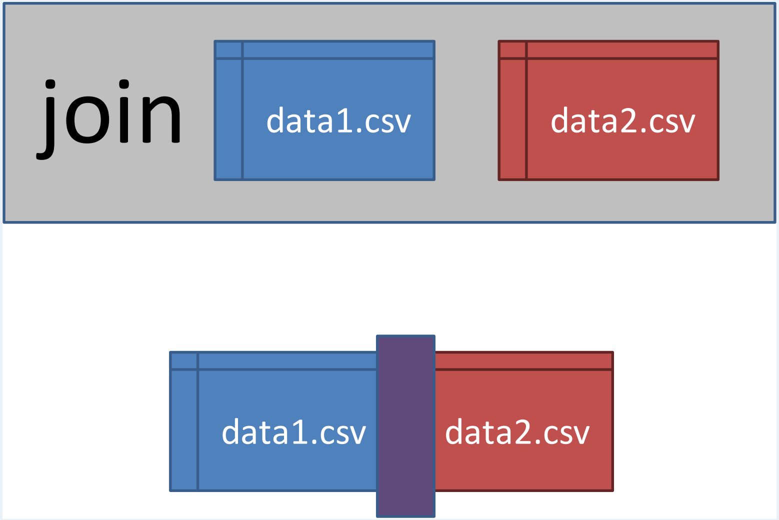
Cat: Stacking Files



Stacking Files with cat

```
cat pop.csv founded.csv > stacked.csv
cat stacked.csv
```

Join



Joining with join

```
join -t, pop.csv founded.csv > joined.csv
```

Summarizing Data

Counts

```
wc diamonds.csv
wc -l diamonds.csv
wc -l diamonds.csv | sed 's/ .*//'
```

Unique Observations

```
sort -u diamonds.csv | wc -l
```

```
tot=`wc -l diamonds.csv | sed 's/ .*//'`
unq=`sort -u diamonds.csv | wc -l`
echo $(($tot - $unq))
```

Basic Variable Operations

```
awk -F '\t' '{ sum += $5 } END { print sum }' diamonds.tsv
awk -F ',' '{ sum += $5 } END { print sum }' diamonds.csv

awk -F '\t' '{ sum += $5 } END { print sum/NR }' diamonds.tsv
awk -F '\t' '{ sum += $4 } END { print sum/NR }' diamonds.tsv
```

Making a Histogram

```
# histogram
tail -n +2 diamonds.csv | cut -d, -f 2 | sort | uniq -c > hist
cat hist.txt
sort -rn hist.txt -o hist.txt
cat hist.txt



sed -i -e 's/^ *//g' -e 's/ /,/' hist.txt
cat hist.txt
sed -i '1 i\Count,Cut' hist.txt
cat hist.txt
```

Wrapup

Using the Shell for Data Processing and Analysis

- Remarkably useful for simple tasks.
- The more complex the task, less appropriate shell is.
- `sed` and `grep` are standard tools in the field. Learn them!
- Homework.
- Next time: git and GitHub