Lecture 21 - awk and make

DSE 511

Drew Schmidt
2022-11-08

Announcements

- Nothing unresolved from last time
- Homework:
  - Deadline extended to 11:59pm Wed Nov 9
- Hard cutoff today 5:35pm
- Questions?

Content

- awk
- make

awk

awk

- A very powerful tool!
- If you can figure out how to use it…
- Named after its authors Aho, Weinberger, and Kernighan.
- Technically its own DSL

awk

CLI Tool

```
awk 'BEGIN{print "hello world"}'
```

Programs

```
cat hw.awk
```

```
BEGIN {
  print "hello world"
}
```

```
awk -f hw.awk
```

```
hello world
```

awk

```
awk '$0 ~ /,TYS,/' ~/sw/data/airlines/csv/1987.csv | head
```

```
## 1987,10,1,4,614,615,559,552,UA,282,NA,45,37,NA,7,-1,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,2,5,618,615,559,552,UA,282,NA,41,37,NA,7,3,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,3,6,621,615,602,552,UA,282,NA,41,37,NA,10,6,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,4,7,614,615,550,552,UA,282,NA,36,37,NA,-2,-1,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,5,1,614,615,549,552,UA,282,NA,35,37,NA,-3,-1,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,6,2,613,615,552,552,UA,282,NA,39,37,NA,0,-2,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,7,3,614,615,552,552,UA,282,NA,38,37,NA,0,-1,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,8,4,614,615,553,552,UA,282,NA,39,37,NA,1,-1,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,9,5,614,615,550,552,UA,282,NA,36,37,NA,-2,-1,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
## 1987,10,10,6,613,615,550,552,UA,282,NA,37,37,NA,-2,-2,TYS,BNA,152,NA,NA,0,NA,0,NA,NA,NA,NA,NA
```

awk

```
awk 'BEGIN{FS=",";OFS="\t"}{$1=$1; print}' ~/sw/data/airlines/csv/1987.csv | head
```

```
## Year    Month   DayofMonth      DayOfWeek       DepTime CRSDepTime      ArrTime CRSArrTime      Uni
## 1987    10      14      3       741     730     912     849     PS      1451    NA      91      79      NA      23
## 1987    10      15      4       729     730     903     849     PS      1451    NA      94      79      NA      14
## 1987    10      17      6       741     730     918     849     PS      1451    NA      97      79      NA      29
## 1987    10      18      7       729     730     847     849     PS      1451    NA      78      79      NA      -2
## 1987    10      19      1       749     730     922     849     PS      1451    NA      93      79      NA      33
## 1987    10      21      3       728     730     848     849     PS      1451    NA      80      79      NA      -1
## 1987    10      22      4       728     730     852     849     PS      1451    NA      84      79      NA      3       -
## 1987    10      23      5       731     730     902     849     PS      1451    NA      91      79      NA      13
## 1987    10      24      6       744     730     908     849     PS      1451    NA      84      79      NA      19
```

awk

```
awk 'BEGIN{FS=",";OFS="\t"}{$1=$1;print $17, $18}' ~/sw/data/airlines/csv/1987.csv | head
```

```
## Origin    Dest
## SAN    SFO
## SAN    SFO
## SAN    SFO
## SAN    SFO
## SAN    SFO
## SAN    SFO
## SAN    SFO
## SAN    SFO
## SAN    SFO
```

awk

```
awk 'BEGIN{FS=",";OFS="\t"}{$1=$1;print $17, $18}' ~/sw/data/airlines/csv/1987.csv | \
  grep TYS | head
```

```
## TYS     BNA
## TYS     BNA
## TYS     BNA
## TYS     BNA
## TYS     BNA
## TYS     BNA
## TYS     BNA
## TYS     BNA
## TYS     BNA
## TYS     BNA
```

awk

```
awk 'BEGIN{FS=",";OFS="\t"}{$1=$1;print $17, $18}' ~/sw/data/airlines/csv/1987.csv | \
  grep TYS | sort | uniq -c
```

```
##     631 ATL    TYS
##     480 BNA    TYS
##     351 CLT    TYS
##      89 CVG    TYS
##     422 MEM    TYS
##      59 ORD    TYS
##     265 PIT    TYS
##      86 TRI    TYS
##     720 TYS    ATL
##     393 TYS    BNA
##     353 TYS    CLT
##      90 TYS    CVG
##     343 TYS    MEM
##      61 TYS    ORD
##     360 TYS    PIT
##      92 TYS    TRI
```

awk

- Can do A LOT more
- We're not going to show that off
- My take:
  - Can be helpful for some quick CLI operations
  - For data science: usually better off using R or Pandas
- Where to learn more:
  - AWK https://en.wikipedia.org/wiki/AWK
  - The GNU AWK (gawk) User's Guide https://www.gnu.org/software/gawk/manual/gawk.html

make

make

- "[A] build automation tool" - Wikipedia
- A workflow tool
- Often reserved for compiled languages
  - C
  - Fortran
  - ...
- Has many uses though!

Example

```
hello:
    Rscript -e "cat('hello world\n')"
```

```
make
```

```
Rscript -e "cat('hello world\n')"
hello world
```

Some Rules

- A makefile is made up of *rules*
- A rule is made up of *targets*, *pre-requisites*, and *recipes*
- Rules should go into `Makefile`
  - Must be named *exactly* `Makefile`
  - Or specify the file name `make -f mymakefile`
- All rules must look like this

```
target: pre-requisite
TAB recipe
```

- A recipe can be basically anything
- To make a specific target type `make target_name`
- The first line will automatically run if you just type `make`

A Quote (Paraphrasing)

> Makefiles are the sourdough starters of software.

Jenny Bryan (I think?)

Example

```
export SHELL=/usr/bin/bash
all: print_results

write:
    Rscript -e "write.csv(iris, file = 'iris.csv')"

extract: write
    awk 'BEGIN{FS=","}{if(NR!=1){print $$6}}' iris.csv > col.txt

summarize: extract
    cat col.txt | sort | uniq -c > summary.txt

print_results: summarize
    cat summary.txt

clean:
    rm -rf iris.csv summary.txt col.txt
```

Example

```
make
```

```
Rscript -e "write.csv(iris, file = 'iris.csv')"
awk 'BEGIN{FS=","}{if(NR!=1){print $6}}' iris.csv > col.txt
cat col.txt | sort | uniq -c > summary.txt
cat summary.txt
     50 "setosa"
     50 "versicolor"
     50 "virginica"
```

```
ls
```

```
col.txt  iris.csv  Makefile  summary.txt
```

```
make clean && ls
```

```
rm -rf iris.csv summary.txt col.txt
Makefile
```

Some Observations

- Each rule is like a little function
- Rules probably shouldn't be *overly* complicated
- `make` automatically resolves the execution order, once pre-requisites are specified
- We're not yet taking advantage of one of its best features...

(Re-)Generating Targets

- `make` can automatically check if a file needs to be re-generated
- Based on timestamps (if output older than modification...)
- One of its best features!
- But you have to name things "correctly"
  - An output is a target is an output
- Can force rebuilds with `.PHONY`
  - `.PHONY: all clean`

Example

```
export SHELL=/usr/bin/bash
all: print_results

iris.csv:
    Rscript -e "write.csv(iris, file = 'iris.csv')"

col.txt: iris.csv
    awk 'BEGIN{FS=","}{if(NR!=1){print $$6}}' iris.csv > col.txt

summary.txt: col.txt
    cat col.txt | sort | uniq -c > summary.txt

print_results: summary.txt
    cat summary.txt

clean:
    rm -rf iris.csv summary.txt col.txt
```

Example

```
make
```

```
Rscript -e "write.csv(iris, file = 'iris.csv')"
awk 'BEGIN{FS=","}{if(NR!=1){print $6}}' iris.csv > col.txt
cat col.txt | sort | uniq -c > summary.txt
cat summary.txt
     50 "setosa"
     50 "versicolor"
     50 "virginica"
```

```
make
```

```
cat summary.txt
     50 "setosa"
     50 "versicolor"
     50 "virginica"
```

Other Features

- Variables
- Sourcing other makefiles
- Wildcards
- Built-in and automatic recursion!
- Can even run things in parallel! (`make -j`)

My Slides Makefile

```
SRCS=$(wildcard *.Rmd)
HTML=$(SRCS:.Rmd=.html )
PDF=$(SRCS:.Rmd=.pdf )

%.html: %.Rmd
	./bin/build.r $<
html: $(HTML)

%.pdf: %.html
	./bin/topdf.r $<
pdf: $(PDF)

all: html

clean:
	rm -rf *.pdf *.html
```

```
make
```

```
# ...
output file: lecture21.knit.md
# ...
Output created: lecture21.html
```

```
make
```

```
make: 'html' is up to date.
```

```
make pdf
```

```
./bin/topdf.r lecture21.html
```

```
make pdf
```

Wrapup

Wrapup

- `awk` is very powerful, but you're probably better off just using R/Python most of the time.
- Your workflow tool probably isn't as good as `make`
- Next time: Shell Scripting (Part 1 of 2)

Questions?