Andy Gomez-Serrano

Prof. Adrianna Holden-Gouveia

CIS-245

February 20, 2025

Awk

1. Print all the First Names.

Type awk ‘{print $1}’ Awklab

• print : displays whatever is being asked (Prints all the 1st names)

• $1 : Is the first field (first names)

• { } : code to be executed (print $1)

• ‘ ‘ : to avoid interpreting any special characters if they were to be introduced

A screen shot of a computer

Description automatically generated

1. Print phone numbers for Tom and Frodo after their names

Type awk -F: ‘$1 ~ “Tom” || $1 ~ “Frodo” {print $1, $2,}' AwkLab

• -F: Since the fields are using colons as delimiters this is where it separate

the fields

• ‘$1 ~ “Tom” || $1 ~ “Frodo”: searches for both Tom and Frodo

• {print $1, $2,}: 1st field is the full name. 2nd field is the phone number

A black screen with white text

AI-generated content may be incorrect.

1. Print Peregrin’s full name and phone number area code only.

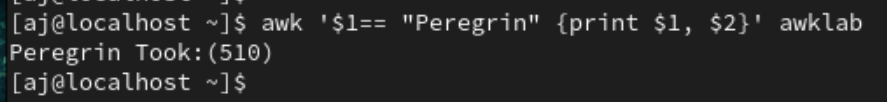
Type awk ‘$1== “Peregrin” {print $1, $2,}' AwkLab

• $1==: Assign the first line to be Peregron

• “Peregrin”: searches for Peregrin

• {print $1, $2}: displays the 1st field which is the 1st name. displays the 2nd

field, which is the last name, display the 3rd field which is the area code



4. Print all phone numbers (full number) in the 123-area code along with

the names

Awk -F: ‘$2 ~ /^\(123\)/ {print $1, $2}’ awklab

* -F':' Sets the field separator
* $2 ~ /\(123\)/: Checks if the second field (phone number) contains "(123)", which indicates the 123 area code.
* {print $1 ": " $2}: Prints:
* $1: The name.
* $2: The full phone number.

A black background with white text

Description automatically generated

5. Print all Last names beginning with either a T or D (careful of middle

names!)

awk ‘{if ($2 ~ /[TD]/) print $0}’ awklab

* $2 ~ /^[TD]: Checks if the second field ($2, which should be the last name) starts with 'T' or 'D' (^ denotes the beginning of the string).
* {print $2}: Prints only the last name.

A screen shot of a computer

Description automatically generated

6. Print all first names containing four or less characters.

Awk ‘length($1) <= 4 {print $1}’ awklab

* Awk: Invokes the awk command-line tool for text processing.
* length ($1) <= 4: length($1) calculates the length of the first field ($1). The condition <= 4 ensures that only words with 4 or fewer characters are processed.
* {print $1}: Prints the first field ($1), which is typically the first name in a structured dataset

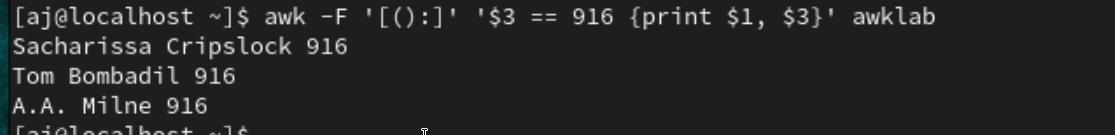
A black screen with white text

Description automatically generated

7. Print the first names and area codes of all those in the 916-area code.

Awk -F ‘[():]’ ‘$3 == 916 {print $1, $3}’ awklab

* -F '[():]': This effectively isolates the area code from the phone numbers.
* $3 == 916: Checks if the third field ($3) is equal to 916, meaning the area code is 916.
* {print $1, $3}: Prints the first field ($1), which is the name. Prints the third field ($3), which is the area code (916)



8. Print Sacharissa’s campaign contributions following her name. Each

value should be printed with a leading dollar sign, e.g., $250 $100 $175.

Awk -F’:’ ‘/Sacharissa/ {print “%s”, $1; for (i=3; i<=NF; i++) print “$s”, $1; print “”}’ awklab

1. awk -F':'

* awk is a command-line text-processing tool.
* -F':' tells awk to use : (colon) as the field separator

2. '/Sacharissa/'

* This part searches for lines in awklab that contain the string "Sacharissa".
* If a line contains this string, the actions inside { ... } will be executed.

3. {print “%s”, $1;

* print "%s", $1; attempts to print the first field ($1), but:
  + The "%s" format specifier is incorrect in awk. awk does not use printf inside print.
  + Instead, it should be print $1; or use printf properly.

4. for (i=3; i<=NF; i++) print “$s”, $1;

* NF is the number of fields in the line.
* This loop starts at field 3 ($3) and iterates through all remaining fields.
* Problem: "$s" is incorrect syntax in awk. The intended behavior is unclear, but it looks like the user wants to print something in a loop.

5. print “”

* This prints an empty line for spacing.

A black screen with white text

Description automatically generated

9. Print last names followed by a comma and the phone number. Be

careful of the last name’s format.

Awk -F’[: ()]’ ‘{split($1, name, “ “); print name [length(name)] “,”$3,$4}’ awklab

* -F’[()]’ : the colon and parentheses separate the fields
* {split($1, name, “ ”)} : splits the 1st field which is the name into separate fields
* Print name[length(name)]”,”$3,$4) : displays the last element of the name array, followed by a comma, and then the 3rd and 4th field which is the phone number.

A screenshot of a computer

Description automatically generated

10. Print the first and last names of those who contributed more than $110

in the last month. Make sure to include their last month contribution

amount after the name

Type awk -F':' '{if ($5 > 110) print $1, $5}' AwkLab

* + -F’:’ : the colon and parentheses separate the fields
  + {if ($5 > 110) print $1, $5} - since we are using the if statement, the 5th field is greater than 110, we want it to display the 1st field (name) and the 5th field (last contribution)

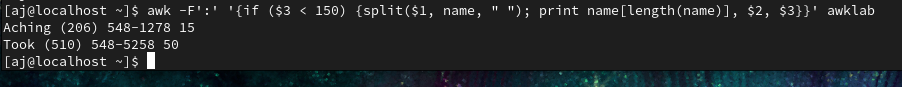
A screenshot of a computer

Description automatically generated

11. Print the last names, phone numbers, and first month contribution of

those who contributed less than $150 in the first month.

Type awk -F':' '{if ($3 < 150) {split($1, name, " "); print name[length(name)], $2, $3}}' AwkLab

* -F’:’ : the colon and parentheses separate the fields
* {if($3 < 150) - looks for numbers under 150 in the 3rd field
* • {split($1, name, “ ”)} : splits the 1st field which is the name
* print name[length(name)], $2, $3}} - displays the last name followed by the 2nd and 3rd fields. 

12. Print the first names and contribution of those who contributed between $10 and $200 in the first month.

Type awk -F':' '{if ($3 >= 10 && $3 <= 200) {split($1, name, " "); print name[1], $3}}' AwkLab

* -F’:’ : the colon and parentheses separate the fields
* {if ($3 >= 10 && $3 <= 200) - Checks if the third field ($3) is between 10 and 200.
* {split($1, name, " "); - splits the 1st field put into an array named name and uses a space as a delimiter by using the quotations with a space between them.
* print name[1], $3}}' - displays the 1st field (first name) and the 3rd field.

A screen shot of a computer

Description automatically generated

13. Print the first name, last names and total contributions of those who

contributed less than $700 over the three-month period.

Awk -F’:’ ‘{if ($5 < 700) print $1, $5}’ awk lab

* -F':': This means awk will split each line based on : and treat each part as a separate field.
* if ($5 < 700): Checks if the fifth field ($5) is less than 700. $5 is assumed to be a numeric value (e.g., a contribution amount).
* print $1, $5: Prints the first field ($1) → typically a name. Prints the fifth field ($5) → assumed to be a numeric value (e.g., contribution amount).

A screenshot of a computer

Description automatically generated

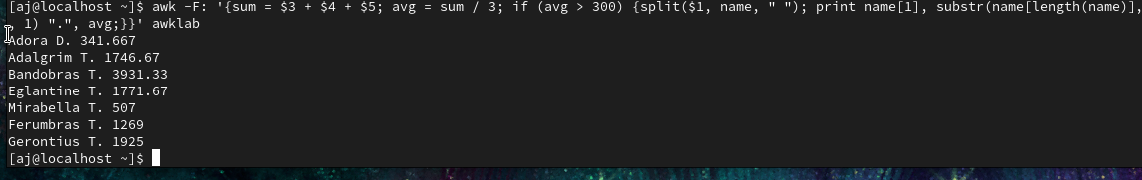
14. Print the first names and first letter of the last name, and average

contribution of those who had an average contribution of more then

$300

Awk -F: ‘{sum= $3 + $4 + $5; avg = sum / 3; if (avg > 300) {split($1, name, “ “); print name[1], substr(name[length(name)],1) “.”, avg;}}’

* -F: Sets colons (:) and spaces ( ) as field separators to correctly extract the first and last names.
* avg = ($3 + $4 + $5) / 3: Calculates the average contribution using the last three numeric fields.
* if (avg > 300): Checks if the average contribution is greater than 300.
* $1: First name.
* Avg: The calculated average contribution.



15. Print the last name and area code of those not in the 916-area code.

16. Print each record preceded by the number of the record.

Awk ‘{print NR “:” $0}’ awklab

* NR is a variable that keeps track of the current record number.
* $0 represents the entire line (record).
* print NR ": " $0 prints the record number followed by the full record.

A screenshot of a computer

Description automatically generated

17. Print the name and total contribution of each person.

* -F: sets the field separator
* $3, $4, and $5 represent the three numerical fields.
* total = $3 + $4 + $5 calculates the sum of these fields.
* print $1 ": " total prints the person's name ($1) followed by their total contribution.

A screenshot of a computer program

Description automatically generated

18. Add $10 to Tiffany Aching’s first contribution and print her full name

and first contribution.

19. Change Samwise Gamgee’s name to Sean Astin

Awk ‘{gsub(“Samwise Gamgee”, “Sean Astin”); print}’ awklab

* gsub("Samwise Gamgee", "Sean Astin" ) searches for Samwise Gamgee in the first field (name) and replaces it with Sean Astin.
* Print the new list with the updated name.

A computer screen shot of a computer

Description automatically generated

20. Write an awk script to do the following (MUST be an awk script not

just a bash script or commands on the command line)

(a) Prints the first name of the all the Tooks followed by their total campaign contributions .

(b) Print the full names and contributions of anyone who contributed

between $10 and $200 in the last contribution

(c) Prints the full names and average contribution of those who contributed less than $300 on average