Gabriel Duffy

Module 2 Assignment

ADS-550B

3/7/2024

1.) The Exercise 1 Dataset (located in your assignment prompt in Blackboard) contains a portion of the data from NYC about causes of death for the year 2010. Dataset format: CSV Field names (in order): Year, Ethnicity, Sex, Cause of Death, Death Count. Answer the following questions from this data using UNIX commands.

Commands Used/Code Output and step-by-step explanation

1.1a) How many male record groups does the data have?

```
gabed@DESKTOP-O995O42 ~/tmp
$ grep "Male" deaths.csv | wc -l
163
```

The grep command will search the deaths.csv data for all the cell values = to "Male". The word count command with give me an output in numerical format for all the cell values =" Male". The answer 163 means there are 163 cell values in the deaths.csv data set that have the male value in them.

1.1b) How many Female record groups does the data have?

```
gabed@DESKTOP-O995O42 ~/tmp
$ grep "Female" deaths.csv | wc -l
161
```

The grep command will search the deaths.csv data for all the cell values = to "Female". The word count command with give me an output in numerical format for all the cell values =" Female". The answer 163 means there are 163 cell values in the deaths.csv data set that have the male value in them.

1.2a) How many white female groups are there?

```
gabed@DESKTOP-O995O42 ~/tmp
$ grep -E '^2010.*White.*Female' deaths.csv | wc -l
36
```

The grep command will search the deaths.csv data and extract the records that meet the following requirements: 2010, White, and Female. The –E will allow the grep command to account for more variables in its search. The wc –1 will keep count of the records that match by grep command and will supply those values to me. The answer 36 is how many white females cell values that are in the deaths.csv dataset.

1.2b) Copy entire records of females to a new text file where the records are organized by death count in descending order.

```
gabed@DESKTOP-O995O42 ~/tmp
$  grep "Female" deaths.csv > females_records.csv
```

First, I used the grep commands functionality to extract female cell values from the deaths.csv dataset then I transferred them to a new file that was created to only contain female records.

```
gabed@DESKTOP-O995O42 ~/tmp
$ sort -t ',' -k 5,5nr -o females_records_sorted.csv females_records.csv
```

Second, used the sort of command to sort records by death count located in the fifth column. I then created a final csv file named females_records_sorted to store my final dataset results to this question in a csv file.

```
gabed@DESKTOP-O995O42 ~/tmp
$ cat females_records_sorted.csv
2010,White,Female,Diseases of Heart,5351
2010,White,Female,Malignant Neoplasms (cancer),3438
2010,Black,Female,Diseases of Heart,2282
```

Then, used the cat command to make sure the file is sorted by descending death count and contains only female values.

```
gabed@DESKTOP-O995O42 ~/tmp
$ cp females_records_sorted.csv females_records_final.txt
```

Finally, with my csv file meeting question requirements I had to transition my file to a text format

1.3) What are the three most frequent causes of death for black males?

```
gabed@DESKTOP-O995O42 ~/tmp
$ awk -F, '$2 == "Black" && $3 == "Male" {count[$4]+=$5} END {for (cause in count) print count[cause], cause}' deaths.csv | sort -k1,1nr | head -3
2015 Diseases of Heart
1540 Malignant Neoplasms (cancer)
299 Assault (Homicide)
```

1.) The awk -F command lets Cygwin know deaths.csv data set is a comma separated file, the '$2 == "Black"- lets Cygwin know I'm looking for all the Black cell values in the second column of my data set
2.) The && $3 =="Male" lets Cygwin know I'm looking for all the Male cell values in my third column along with searching the second column for Black cell values,
3.) The {count[$4]+=5} lets Cygwin know I'm looking for the causes of death associated with Black Males along with the number value associated in the fifth column with the diseases in the fourth column
4.) The End { for (cause in count) print count[cause], cause}' deaths.csv lets Cygwin know to tally up all the deaths for each Black Male by Cause of Death and process all records for me to have at the end
5.) The | sort-k1,1nr | this command tells Cygwin I want all the Causes of Death for Black Males sorted by highest to lowest reversed order based of the cell values totals of Death Count
6.) The | head –3 command is telling Cygwin I want the three top lines associated with the highest death count by disease for black males
Answer = Diseases of Heart, Malignant Neoplasms (cancer), and Assault (Homicide) are the three most frequent causes of death for black males.

1.3) Five least frequent causes of death for Hispanic females?

```
gabed@DESKTOP-O995O42 ~/tmp
$ awk -F, '$2 == "Hispanic" && $3 == "Female" {count[$4]+=$5} END {for (cause in count) print count[cause], cause}' deaths.csv | sort -k1,1n | head -5
1 All Censored Causes
1 Anemias
2 Pneumonitis due to Solids and Liquids
2 Tuberculosis (TB)
4 Complications of Medical and Surgical Care
```

1.) The awk –F command lets Cygwin know deaths.csv data set is a comma separated file, the '$2 == "Hispanic"- lets Cygwin know I'm looking for all the Hispanic cell values in the second column of my data set,

2.) The && $3 ==”Female” lets Cygwin know I’m looking for all the Female cell values in my third column along with searching the second column for Hispanic cell values
3.) The {count[$4]+=5} lets Cygwin know I’m looking for the causes of death associated with Hispanic Females along with the number value associated in the fifth column with the diseases in the fourth column
4.) The End { for (cause in count) print count[cause], cause}’ deaths.csv lets Cygwin know to tally up all the deaths for each Hispanic Females by Cause of Death and process all records for me to have at the end
5.) The | sort-k1,1n | this command tells Cygwin I want all the Causes of Death for Hispanic Females sorted by highest to lowest based of the cell values totals of Death Count
6.) The | head –5 command is telling Cygwin I want the top five lines associated with the lowest death count by disease for Hispanic females
Answer = All Censored Causes, Anemias, Pneumonitis due to Solids and Liquids, Tuberculosis (TB), Complications of Medical and Surgical Care are the five least frequent cause of death for Hispanic females. (Zen et al.,2021)

2.) Obtained from UNICEF, the Exercise 2 Dataset (located in your assignment prompt in Blackboard) contains data related to the population of 70+ countries for the year 2017. Dataset format: CSV Field names (in order): Country, Population, Urban Population, Percentage of Urban Population. Answer the following questions from this data using UNIX commands:

2.1) Which country has the lowest percentage of urban population?

```
gabed@DESKTOP-O995O42 ~/tmp
$ awk -F, '{print $4, $1}' population.csv | sort -n | head -1
 Burundi
```

1.) The –F command lets Cygwin know population.csv data set is a comma separated file
2.) The ‘{print $4, $1}’ population.csv command is telling Cygwin I want the results printed by the fourth column Percentage of Urban Population by the first column Country and population.csv is that data set I want this command ran on
3.) The | sort –n | command is telling Cygwin I want the data set ordered by lowest percentage of urban population and country
4.) The head –1 command is telling Cygwin I want the first line of the country with the lowest urban population as my output
Answer = Burundi has the lowest percentage of urban population of any country in the population.csv dataset

2.2) List the countries where the urban population is more than 10 million and yet they comprise less than half of the population.

```
gabed@DESKTOP-O995O42 ~/tmp
$ awk -F, ' $3 > 10000000 && $4 < 50 { print $1}' population.csv ~
Bangladesh
Democratic Republic of the Congo
Egypt
Ethiopia
India
Kenya
Myanmar
Pakistan
Philippines
Sudan
Thailand
United Republic of Tanzania
Viet Nam
Yemen
```

1.) The –F command lets Cygwin know population.csv data set is a comma separated file
2.) The ‘ $3 > 10000000 command lets Cygwin know that I’m looking for an urban population over 10 million
3.) The && $4<50 command says to Cygwin along with the urban population having to be over 10 million I also want the percentage of the Urban population to be less than 50 percent
4.) The { print $1} population.csv command show Cygwin I want the countries names that have a population more than 10 million with an urban population less than 50 percent as my output
Answer = The output lists 14 countries that have an urban population over 10 million but of that 10 million less 50% of people live in a urban setting

3.) For the following exercise, use the Exercise 3 Dataset (located in your assignment prompt in Blackboard), which contains availability of essential medicines in 38 countries for the years 2007 - 2013, obtained from the World Health Organization (WHO). Dataset format: CSV Field names (in order): Country, Median availability of selected generic medicines (%) - Private, Median availability of selected generic medicines (%) - Public

3.1) Which country had the lowest percentage median availability of selected generic medicines in private?

```
gabed@DESKTOP-O995O42 ~/tmp
$ awk -F'\t' 'NR > 1 && ($2 < min || min == "") {min = $2; country = $1} END {print country}' medicines.txt
India
```

Median availability of elected generic medicines in private sector was column 2 of the medicines csv. File which I converted to a text file because no matter what I command I tried in csv format I could not pull the information I needed in. Country names were in column 1. I made sure that my output took both into account when printing my result. India has the lowest percentage median availability of selected generic medicines in the private sector.

3.2) List the top five countries with the highest percentage of public and private median availability of selected medicines in 2007–2013

```
gabed@DESKTOP-O995O42 ~/tmp
$ awk -F'\t' 'NR > 1 {public[$3]+=$2; private[$2]+=$2} END {printf "Top 5 countries with the highest public availability:\n"; PROCINFO["sorted_in"]="@val_num_desc"; count = 0; for (i in public) { if (length
i) > 0) { print i, public[i]; count++; if (count == 5) break; } } printf "\nTop 5 countries with the highest private availability:\n"; PROCINFO["sorted_in"]="@val_num_desc"; count = 0; for (j in private) {
f (length(j) > 0) { print j, private[j]; count++; if (count == 5) break; } } }' medicines.txt
Top 5 countries with the highest public availability:
 167
 133.3
 98.2
 941
 91.7

Top 5 countries with the highest private availability:
86.7 173.4
70 140
50 100
100 100
98.2 98.2
```

(Spiakloo et al.,2020) (Zen et al.,2021). Used these resources for command references that I could implement in my code writing

The top five countries with highest percentage of public median availability are Ukraine, Cook Islands, Russia, Oman, and Iran.

The top five countries with highest percentage of private median availability are Syrian Arab Republic, Sudan, Russia, Iran, and Afghanistan.

3.3) List the top three countries where it is best to rely on the private availability of selected generic medicines than public. Explain your answer with valid reasons.

```
gabed@DESKTOP-O995O42 ~/tmp
$ awk -F'\t' 'NR>1{p[$3]+=$2}END{for(i in p)print i,p[i]|"sort -k2n|head -3"}' medicines.txt
 13.3
 21.7
 22.2
```

(Spiakloo et al.,2020) (Zen et al.,2021). Used these resources for command references that I could implement in my code writing

I set up my command a little differently than the question states, I wanted the command to show me the countries with lowest median availability of generic medicines in public sector than I would be able to recommend those countries to rely on private availability. Thet top three countries that should rely on private availability are, Brazil, China, and Hati.

4.) In this exercise, we practice writing Python functions and loops.

Python Commands in Jupyter Notebook

4.1) Define a Python function is_in_high_school(age) that takes the age of a person and determines if the person is in high school or not. Assume that for a person to be in high school, their age should be between 14 and 18, inclusive.

```
In [4]: def is_in_highschool(age):
            return 14 <= age < 18
```

I first defined my function in Jupyter with the Python command def is_in_highscool then gave the function a parameter (age) then a ran a return statement on the age of highschoolers.

4.2) Iterate over the list [25,18,9,13,34,15,22,17,12,37,15] and call your function is_in_high_school(age) on each age value.

In [6]:
```
ages = [25, 18, 9, 13, 34, 15, 22, 17, 12, 37, 15]

for age in ages:
    if is_in_highschool(age):
        print(f"{age} in high school.")
    else:
        print(f"{age} not in high school.")
```

```
25 not in high school.
18 not in high school.
9 not in high school.
13 not in high school.
34 not in high school.
15 in high school.
22 not in high school.
17 in high school.
12 not in high school.
37 not in high school.
15 in high school.
```

First thing I did was supply Jupyter with the list of ages I want iterated, then I wanted to specify which variable I wanted the numbers iterated to, then I gave an if command to print the results with text so I can quickly see which ages are in high school and which ones are not.

4.3) Calculate the percentage of people in your age list not going to high school.

In [8]:
```
ages = [25, 18, 9, 13, 34, 15, 22, 17, 12, 37, 15]

# Count of non highschoolers
not_in_highschool_count = sum(1 for age in ages if not is_in_highschool(age))

# Calculate the percentage
percentage_not_in_highschool = (not_in_highschool_count / len(ages)) * 100

print(f"The percentage of non highschoolers: {percentage_not_in_highschool:.2f}%")
```

```
The percentage of non highschoolers: 72.73%
```

I supplied the list of ages, then ran a count on non-high school ages then calculated the percentage by dividing the count of non-highschoolers count by the lens command and times by 100 to get a clean number that can easily be converted to a percent., then I used the print command to print the percentage of people not in high school which 72.73%.

References:

1.) Zen, P. (2021). AWK command in Unix | AWK tutorial for scripting | awk shell scripting tutorial | UNIX | linux. Pavan the ZEN. https://www.youtube.com/watch?v=kIEKVT1OvHU2

2.) Edureka. (2021). Basic UNIX Commands | UNIX Shell Commands Tutorial for Beginners | UNIX Training | Edureka. Edureka https://www.youtube.com/watch?v=-OXftCGd4_I

3.) Spiakloo.(2020). 47 Useful UNIX Commands for Data Science. Spiakloo. https://www.youtube.com/watch?v=-OXftCGd4_I

4.) Lab Session 2.1: Introduction to Linux Command Line. University of San Diego ADS-500B. https://sandiego.instructure.com/courses/7146/pages/lab-session-2-dot-1-introduction-to-linux-command-line?module_item_id=4223015

5.) Elder, R. (2019). **An Introduction To Data Science On The Linux Command Line. Robert Elder Software.** https://blog.robertelder.org/data-science-linux-command-line/