CIT 371 Lab 9: Regular expressions and grep (egrep)

Start Coivcenter and your VM, log in as Student.

1. We will start by exploring some egrep commands. These are covered in the chapter 6 PowerPoint slides. There are no questions to answer in this step. su to root (password is cit371) and cd to /etc. Enter each of the following commands. You will have to see if you can identify the meaning behind each regex.
   1. **egrep ‘[0-9]+’ \***
   2. **egrep ‘[A-Z]+{12,}’ \***
   3. **egrep ‘[A-Z]+=[0-9]+’ \***
   4. **egrep ‘#$’ \*** Notice that the output doesn’t give you much useful information, repeat this adding the option **–l** to egrep. The **–l** option prints only the matching files, not the lines of each match.
   5. **egrep ‘^$’ \*** Again, repeat this instruction using **–l.**
   6. **egrep ‘[A-Z]+=’ bashrc**
   7. **egrep ‘if \[‘ bashrc**
   8. **egrep ‘\(\) \{‘ bashrc**
   9. **egrep ‘[0-255].[0-255].[0-255].[0-255]’ \***
   10. In the previous example, we wanted to list from files any that had IP addresses. You might notice in the output that you got a lot of output that consists of non-IP addresses. Can you figure out what is wrong with the above regex? Use this regex instead:

**[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}**

Can you figure out why this regex is more appropriate?

**This defines how many numbers and literal characters you want in order to find all just IP addresses.**

Exit as root.

You are responsible for answering the questions from step 2. cd to ~/FILES and look at the sales.txt. You will use grep/egrep on this file.

1. Use egrep commands to search for patterns in the sales.txt file.
   1. In sales.txt, we will search for all lines that contain the month Feb. The command is

**egrep ‘Feb’ sales.txt**

*What was the response? Now how would you search for all lines that contain an entry for Smith?*

***The results were a list of the month Feb with names such as Barber, Dean, and Smith. In order to look for all lines that contain smith we need to use the command egrep ‘Smith’ sales.txt .***

* 1. The –c option for egrep counts the number of occurrences. Write egrep commands to count the number of entries in the file that contain Cameron. *What command did you enter? How many were found?*
     1. **The command I used was egrep -c ‘Cameron’ sales.txt. The count was 2.**
  2. *Repeat b for entries that contain KY.* 
     1. ***4***
  3. Let’s find all lines that contain a commission rate of .15. Enter the command

**egrep ‘.15’ sales.txt**

Look at the response. *What entries appeared that shouldn’t? Why did they appear? How will you fix this regex?* Fix it and try again to see if you have the correct answer.

**The entries that appeared were that of .15, .16, .10, .15. These appeared because when we used egrep the period . means match any character, so it is matching any character. In order to fix this, we need to use the command egrep ‘\.15’ sales.txt so that the period is a literal character and not a metacharacter.**

* 1. Let’s assume we want to find all records whose Sales value is over 9999. Type the command **egrep ‘[1-9][0-9]{4}’ sales.txt** *Why did we use [1-9] before [0-9]{4} instead of [0-9]{5}?* 
     1. **When I tested both of these they worked the same. But you would use the [1-9] before the before [0-9]{4} instead of [0-9]{5}**, **because we have to define the egrep to say search for a 1-9 in the 5th value to start off the over 10000 followed by any 4 digts to make up the over 9999 value. The 5th value should not be a zero but between a 1-9.**
     2. *Text

        Description automatically generated with medium confidence*
  2. Following up on part e, assume we instead want to find everyone who had less than 10000 for Sales. The command you might think of would be

**egrep ‘[1-9][0-9]{3}’ sales.txt**

Enter this command. You will see that it responded with all entries. *Why didn’t it work? See if you can figure out.*  One solution to solve this problem is to repeat the command from part e but add the –v option to your egrep command. Try it out to see if it worked.

**I believe this command didn’t work out because of the {3} we are only searching for 3 digits that apply no matter how many numbers. So the -v is an invert option that helps us solve this problem.**

* 1. To find all entries of sales in either OH or PA, you can use **egrep ‘OH|PA’ sales.txt**. Try it. Write an egrep command to find all entries that contain either Barber or Cameron. *What command did you enter?*
     1. **Egrep ‘Barber|Cameron’ sales.txt**
  2. Following up on part g, *how could you search for entries that contain both OH and PA?* Hint: you cannot do this with a single egrep statement. *Include your command in your answers.*
     1. **Egrep ‘OH’ sales.txt | egrep ‘PA’**
  3. We want to find all entries that contain at least 4 states. Notice that separating states is a comma and that commas are only used to separate states. Try to figure out an egrep command to find all entries that have at least 4 states. *What command did you come up with?*
     1. **egrep '([[:alpha:]]\*\,){3}[[:alpha:]]\*' sales.txt**

Shut down your VM if desired, log out of the VPN if you are using it, and submit your lab report.