Open Source Tools Assignment 2

Due: Thursday, October 23, 2014 at 11:59pm

Overview

Since we're in the middle of election season here in the US, we'll use open source tools to become armchair political analysts. We'll focus on the senate races where which political party has control hangs in the balance. The website http://www.electoral-vote.com has polling data for each of the senate contests on the page: http://www.electoral-vote.com/evp2014/Senate/senate_polls.html. Each line of the table shows an individual poll taken by a pollster (or "Unopposed" if the seat is uncontested). The poll data includes: the state, the candidate names in each party (Democrat, Republican, or Independent, any of which can be blank depending on the election), the polled percentages, the dates that polls cover, and what polling company took the poll.

Although there is raw CSV data for these polls, they do not contain the names of the candidates and therefore we will use the tools we have learned in this class to turn the web page into CSV, then use a new set of tools (e.g. **grep** and **awk**) to analyze the data.

Questions

Part I

Write a sed script that can be used to convert the raw HTML collected from the above web page into plain text. The raw html for this page resides on the department file servers in `/home/unixtool/data/election14/senate_polls.html`. The output should show the values in the table (excluding headers), each column represented on a separate line (empty columns should result in a blank line. For example, the first row would be output as:

```
Alaska
Mark Begich
43
Dan Sullivan
48


Sep 23
Sep 24
Rasmussen
```

1. Write a single **sed** script to create this output. Here are the actions that need to be taken:
   - Remove all lines before and after the table. The table begins after the line that starts with **<table**... and ends at the line that is **</table>**
   - Remove lines that have the tag **<tr>** or **</tr>**, and lines that contain **"small-state-header"**
   - Remove blank lines
   - Convert each ** ** to a space

- Remove all html tags (e.g. **** or ****)
- Remove whitespace at the beginning and the end of each line
- Remove the `%` from lines that are made up of just numbers a trailing `%`
- Remove the `*` from lines that end with a `*`

2. Write an **awk** script to take the output from **sed** in question 1 and create a CSV (comma separated values) file for the data. The format should look like this:

```
Alabama,None,00,Jeff Sessions,100,,,Jan 01,Jan 01,Unopposed
Alaska,Mark Begich,43,Dan Sullivan,48,,,Sep 23,Sep 24,Rasmussen
Alaska,Mark Begich,41,Dan Sullivan,43,,,Sep 18,Sep 21,PPP
```

Part II

Write pipelines to do the following, using the output from awk above. (you may use grep, egrep, tools from the previous homework, but not awk or sed; each should be a single command or pipeline of commands):

3. Count the number of polls that are from the source "Fox News"
4. Count the number of polls that are from the source "SurveyUSA" or "Rasmussen"
5. Count the number of polls that are not from the source "Fox News"
6. Show the lines that have a poll whose duration is a single day (start = end).
7. Print the names of Republican candidates that have over 50% in any poll.
8. Print the names of Democrats whose first name begins with the same letter as the last name.

Part III

For these questions, you must answer each with a single awk script. No other commands may be used.

9. Print the name of the polling organization that had the poll with the widest margin (difference in Republican and Democrat percentages), excluding Unopposed.
10. Print the state that has the greatest number of polls.
11. Show each polling organization and the number of polls from that organization.
12. Print the average number of polls per state.
13. Print the polling organization that gives the highest average score for Democrats.

Turning in the assignment

Copy the file `/home/unixtool/data/assignment2` into your directory and fill in the answers in the appropriate places in this file. It is important that you start with this file, which will help the graders partially automate testing. Only submit the commands themselves, not the output of the commands. Here is an example:

```
## Assignment 2
## Joni Ernst N53943245

#############################################################
## Question 1 ###############################################
#############################################################
```

```
find . -name myfile -print
################################################################
## Question 2 ##################################################
################################################################

find . -name otherfile -print
```

Make sure each command has been tested! You can use the following tool to verify that your homework is in a valid format:

```
/home/unixtool/bin/check_assignment2 assignment2
```

When you have finished, submit using the homework submission system.