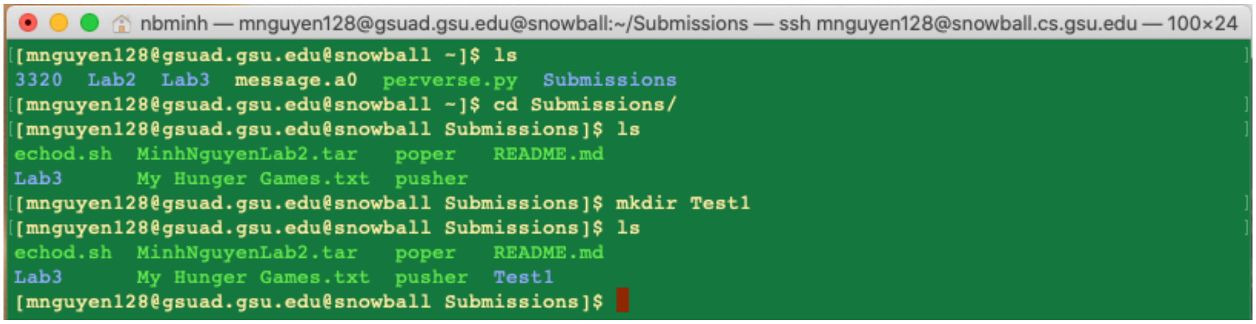
Name: Minh Binh Nguyen
PantherID: 002-46-4288

TEST 1

Part 1

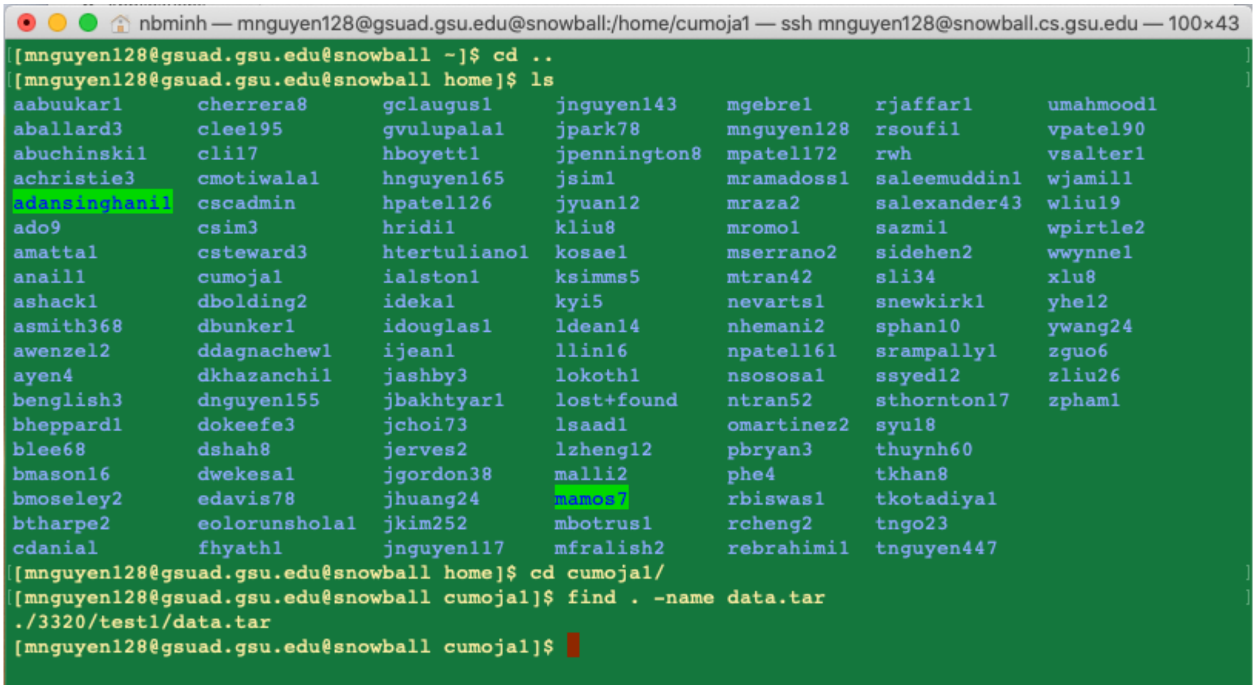
1. I created a directory in my Submissions folder in my home directory and named it “Test1” (Figure 1).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions — ssh mnguyen128@snowball.cs.gsu.edu — 100×24

```
[mnguyen128@gsuad.gsu.edu@snowball ~]$ ls
3320  Lab2  Lab3  message.a0  perverse.py  Submissions
[mnguyen128@gsuad.gsu.edu@snowball ~]$ cd Submissions/
[mnguyen128@gsuad.gsu.edu@snowball Submissions]$ ls
echod.sh  MinhNguyenLab2.tar   poper   README.md
Lab3      My Hunger Games.txt  pusher
[mnguyen128@gsuad.gsu.edu@snowball Submissions]$ mkdir Test1
[mnguyen128@gsuad.gsu.edu@snowball Submissions]$ ls
echod.sh  MinhNguyenLab2.tar   poper   README.md
Lab3      My Hunger Games.txt  pusher  Test1
[mnguyen128@gsuad.gsu.edu@snowball Submissions]$ 
```

Figure 1

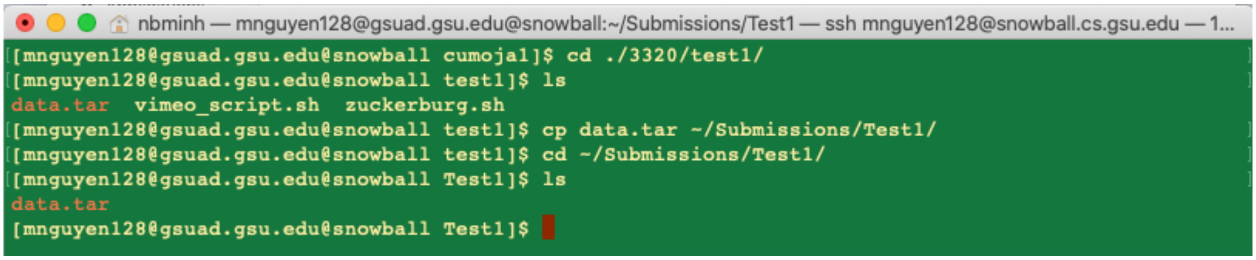
2. Then I used the “find” command to locate the “data.tar” file in /cumoja1 folder (Figure 2).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:/home/cumoja1 — ssh mnguyen128@snowball.cs.gsu.edu — 100×43

```
[mnguyen128@gsuad.gsu.edu@snowball ~]$ cd ..
[mnguyen128@gsuad.gsu.edu@snowball home]$ ls
aabuukar1      cherrera8      gclaugus1     jnguyen143    mgebre1     rjaffar1      umahmood1
aballard3      clee195        gvulupala1    jpark78       mnguyen128  rsoufi1       vpatel90
abuchinski1    cli17          hboyett1      jpennington8  mpatel172   rwh           vsalter1
achristie3     cmotiwala1     hnguyen165    jsim1         mramadoss1  saleemuddin1  wjamil1
adansinghani1  cscadmin       hpatel126     jyuan12       mraza2      salexander43  wliu19
ado9           csim3          hridi1        kliu8         mromo1      sazmi1        wpirtle2
amatta1        csteward3      htertuliano1  kosae1        mserrano2   sidehen2      wwynne1
anail1         cumoja1        ialston1      ksimms5       mtran42     sli34         xlu8
ashack1        dbolding2      ideka1        kyi5          nevarts1    snewkirk1     yhe12
asmith368      dbunker1       idouglas1     ldean14       nhemani2    sphan10       ywang24
awenzel2       ddagnachew1    ijean1        llin16        npatel161   srampally1    zguo6
ayen4          dkhazanchi1    jashby3       lokoth1       nsososa1    ssyed12       zliu26
benglish3      dnguyen155     jbakhtyar1    lost+found    ntran52     sthornton17   zpham1
bheppard1      dokeefe3       jchoi73       lsaad1        omartinez2  syu18
blee68         dshah8         jerves2       lzheng12      pbryan3     thuynh60
bmason16       dwekesa1       jgordon38     malli2        phe4        tkhan8
bmoseley2      edavis78       jhuang24      mamos7        rbiswas1    tkotadiya1
btharpe2       eolorunshola1  jkim252       mbotrus1      rcheng2     tngo23
cdanial        fhyath1        jnguyen117    mfralish2     rebrahimi1  tnguyen447
[mnguyen128@gsuad.gsu.edu@snowball home]$ cd cumoja1/
[mnguyen128@gsuad.gsu.edu@snowball cumoja1]$ find . -name data.tar
./3320/test1/data.tar
[mnguyen128@gsuad.gsu.edu@snowball cumoja1]$ 
```

Figure 2

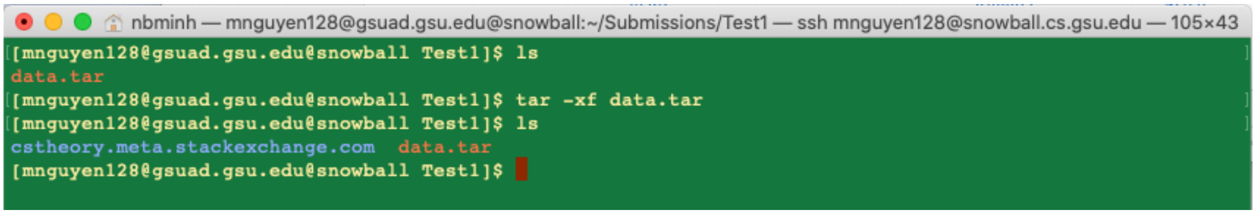
3. I then copied the “data.tar” file into my “Test1” folder (Figure 3).



```
nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 1...
[mnguyen128@gsuad.gsu.edu@snowball cumojal]$ cd ./3320/test1/
[mnguyen128@gsuad.gsu.edu@snowball test1]$ ls
data.tar  vimeo_script.sh  zuckerburg.sh
[mnguyen128@gsuad.gsu.edu@snowball test1]$ cp data.tar ~/Submissions/Test1/
[mnguyen128@gsuad.gsu.edu@snowball test1]$ cd ~/Submissions/Test1/
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ 
```

Figure 3

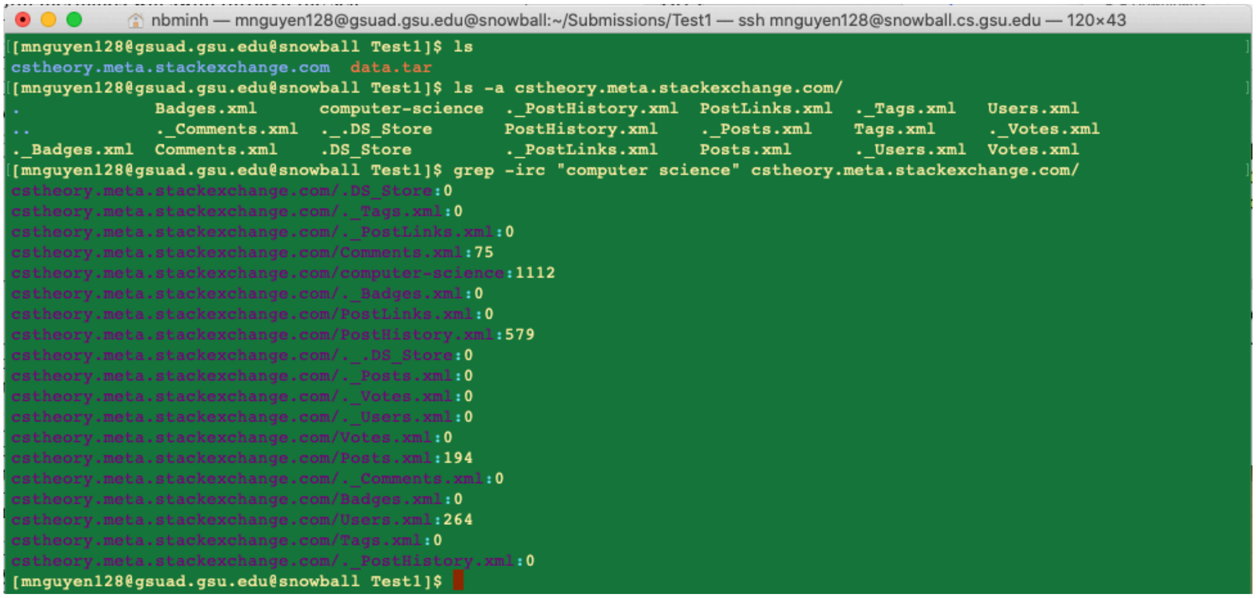
4. Then I extracted all the contents of “data.tar” (Figure 4).



```
nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 105×43
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ tar -xf data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
cstheory.meta.stackexchange.com  data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ 
```

Figure 4

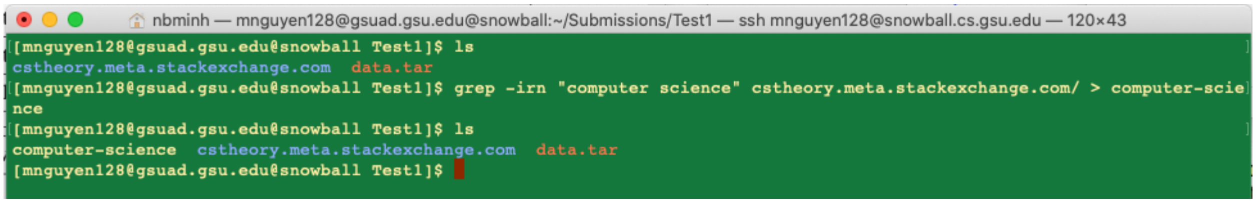
5. I then ran the grep command to count how many lines contain the phrase ”computer science” (ignoring case) (Figure 5).



```
nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 120×43
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
cstheory.meta.stackexchange.com  data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls -a cstheory.meta.stackexchange.com/
.              Badges.xml       computer-science  ._PostHistory.xml  PostLinks.xml  ._Tags.xml   Users.xml
..             ._Comments.xml   ._.DS_Store       PostHistory.xml    ._Posts.xml    Tags.xml     ._Votes.xml
._Badges.xml   Comments.xml     .DS_Store         ._PostLinks.xml    Posts.xml      ._Users.xml  Votes.xml
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ grep -irc "computer science" cstheory.meta.stackexchange.com/
cstheory.meta.stackexchange.com/.DS_Store:0
cstheory.meta.stackexchange.com/._Tags.xml:0
cstheory.meta.stackexchange.com/._PostLinks.xml:0
cstheory.meta.stackexchange.com/Comments.xml:75
cstheory.meta.stackexchange.com/computer-science:1112
cstheory.meta.stackexchange.com/._Badges.xml:0
cstheory.meta.stackexchange.com/PostLinks.xml:0
cstheory.meta.stackexchange.com/PostHistory.xml:579
cstheory.meta.stackexchange.com/._.DS_Store:0
cstheory.meta.stackexchange.com/._Posts.xml:0
cstheory.meta.stackexchange.com/._Votes.xml:0
cstheory.meta.stackexchange.com/._Users.xml:0
cstheory.meta.stackexchange.com/Votes.xml:0
cstheory.meta.stackexchange.com/Posts.xml:194
cstheory.meta.stackexchange.com/._Comments.xml:0
cstheory.meta.stackexchange.com/Badges.xml:0
cstheory.meta.stackexchange.com/Users.xml:264
cstheory.meta.stackexchange.com/Tags.xml:0
cstheory.meta.stackexchange.com/._PostHistory.xml:0
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ 
```

Figure 5

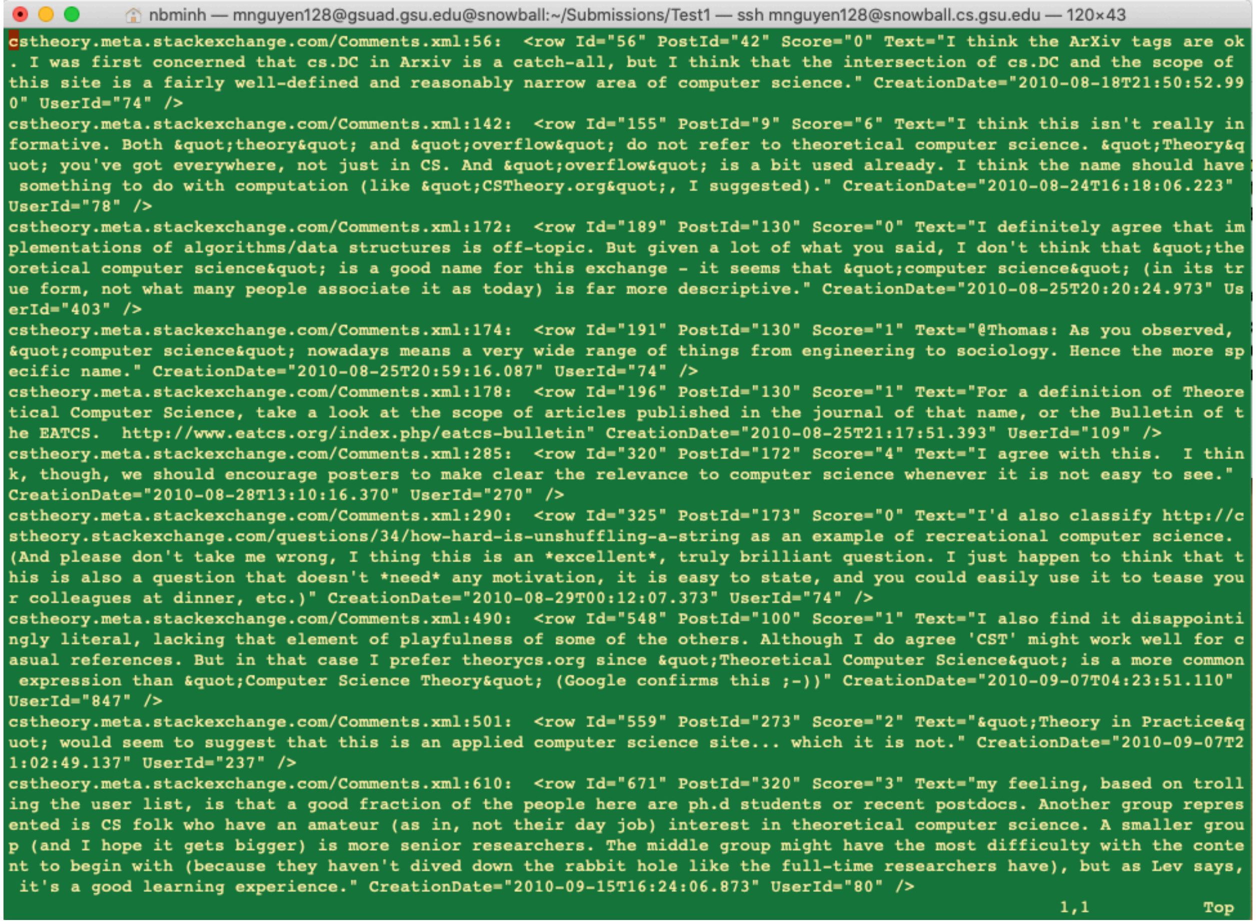
6. I then ran the grep command with “>” to parse the output to a file named “computer-science” (Figure 6).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 120×43

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
cstheory.meta.stackexchange.com  data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ grep -irn "computer science" cstheory.meta.stackexchange.com/ > computer-scie
nce
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
computer-science  cstheory.meta.stackexchange.com  data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ 
```

Figure 6

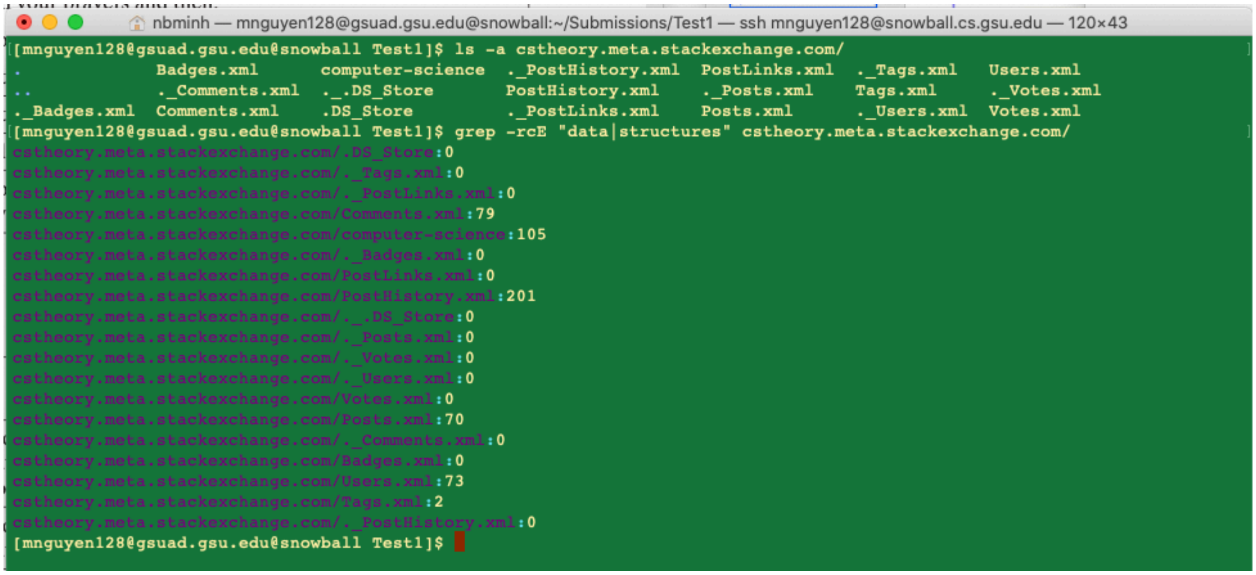
Here is a quick look at the “computer-science” file that I just created (Figure 7).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 120×43

```
cstheory.meta.stackexchange.com/Comments.xml:56:  <row Id="56" PostId="42" Score="0" Text="I think the ArXiv tags are ok
. I was first concerned that cs.DC in Arxiv is a catch-all, but I think that the intersection of cs.DC and the scope of
this site is a fairly well-defined and reasonably narrow area of computer science." CreationDate="2010-08-18T21:50:52.99
0" UserId="74" />
cstheory.meta.stackexchange.com/Comments.xml:142:  <row Id="155" PostId="9" Score="6" Text="I think this isn't really in
formative. Both &quot;theory&quot; and &quot;overflow&quot; do not refer to theoretical computer science. &quot;Theory&q
uot; you've got everywhere, not just in CS. And &quot;overflow&quot; is a bit used already. I think the name should have
 something to do with computation (like &quot;CSTheory.org&quot;, I suggested)." CreationDate="2010-08-24T16:18:06.223"
UserId="78" />
cstheory.meta.stackexchange.com/Comments.xml:172:  <row Id="189" PostId="130" Score="0" Text="I definitely agree that im
plementations of algorithms/data structures is off-topic. But given a lot of what you said, I don't think that &quot;the
oretical computer science&quot; is a good name for this exchange - it seems that &quot;computer science&quot; (in its tr
ue form, not what many people associate it as today) is far more descriptive." CreationDate="2010-08-25T20:20:24.973" Us
erId="403" />
cstheory.meta.stackexchange.com/Comments.xml:174:  <row Id="191" PostId="130" Score="1" Text="@Thomas: As you observed,
&quot;computer science&quot; nowadays means a very wide range of things from engineering to sociology. Hence the more sp
ecific name." CreationDate="2010-08-25T20:59:16.087" UserId="74" />
cstheory.meta.stackexchange.com/Comments.xml:178:  <row Id="196" PostId="130" Score="1" Text="For a definition of Theore
tical Computer Science, take a look at the scope of articles published in the journal of that name, or the Bulletin of t
he EATCS.  http://www.eatcs.org/index.php/eatcs-bulletin" CreationDate="2010-08-25T21:17:51.393" UserId="109" />
cstheory.meta.stackexchange.com/Comments.xml:285:  <row Id="320" PostId="172" Score="4" Text="I agree with this.  I thin
k, though, we should encourage posters to make clear the relevance to computer science whenever it is not easy to see."
CreationDate="2010-08-28T13:10:16.370" UserId="270" />
cstheory.meta.stackexchange.com/Comments.xml:290:  <row Id="325" PostId="173" Score="0" Text="I'd also classify http://c
stheory.stackexchange.com/questions/34/how-hard-is-unshuffling-a-string as an example of recreational computer science.
(And please don't take me wrong, I thing this is an *excellent*, truly brilliant question. I just happen to think that t
his is also a question that doesn't *need* any motivation, it is easy to state, and you could easily use it to tease you
r colleagues at dinner, etc.)" CreationDate="2010-08-29T00:12:07.373" UserId="74" />
cstheory.meta.stackexchange.com/Comments.xml:490:  <row Id="548" PostId="100" Score="1" Text="I also find it disappointi
ngly literal, lacking that element of playfulness of some of the others. Although I do agree 'CST' might work well for c
asual references. But in that case I prefer theorycs.org since &quot;Theoretical Computer Science&quot; is a more common
 expression than &quot;Computer Science Theory&quot; (Google confirms this ;-))" CreationDate="2010-09-07T04:23:51.110"
UserId="847" />
cstheory.meta.stackexchange.com/Comments.xml:501:  <row Id="559" PostId="273" Score="2" Text="&quot;Theory in Practice&q
uot; would seem to suggest that this is an applied computer science site... which it is not." CreationDate="2010-09-07T2
1:02:49.137" UserId="237" />
cstheory.meta.stackexchange.com/Comments.xml:610:  <row Id="671" PostId="320" Score="3" Text="my feeling, based on troll
ing the user list, is that a good fraction of the people here are ph.d students or recent postdocs. Another group repres
ented is CS folk who have an amateur (as in, not their day job) interest in theoretical computer science. A smaller grou
p (and I hope it gets bigger) is more senior researchers. The middle group might have the most difficulty with the conte
nt to begin with (because they haven't dived down the rabbit hole like the full-time researchers have), but as Lev says,
 it's a good learning experience." CreationDate="2010-09-15T16:24:06.873" UserId="80" />
                                                                                                    1,1           Top
```

Figure 7

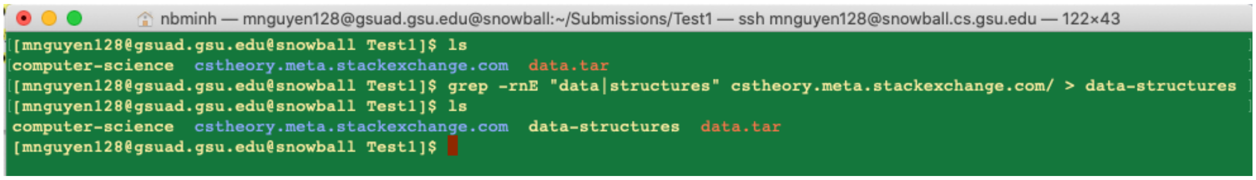
7. I then used a grep command with the powerful flag -E to count all the lines that contain the words “data” or “structures” (exact casing) (Figure 8).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 120×43

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls -a cstheory.meta.stackexchange.com/
.             Badges.xml      computer-science  ._PostHistory.xml  PostLinks.xml  ._Tags.xml   Users.xml
..            ._Comments.xml  ._.DS_Store       PostHistory.xml    ._Posts.xml    Tags.xml     ._Votes.xml
._Badges.xml  Comments.xml    .DS_Store         ._PostLinks.xml    Posts.xml      ._Users.xml  Votes.xml
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ grep -rcE "data|structures" cstheory.meta.stackexchange.com/
cstheory.meta.stackexchange.com/.DS_Store:0
cstheory.meta.stackexchange.com/._Tags.xml:0
cstheory.meta.stackexchange.com/._PostLinks.xml:0
cstheory.meta.stackexchange.com/Comments.xml:79
cstheory.meta.stackexchange.com/computer-science:105
cstheory.meta.stackexchange.com/._Badges.xml:0
cstheory.meta.stackexchange.com/PostLinks.xml:0
cstheory.meta.stackexchange.com/PostHistory.xml:201
cstheory.meta.stackexchange.com/._.DS_Store:0
cstheory.meta.stackexchange.com/._Posts.xml:0
cstheory.meta.stackexchange.com/._Votes.xml:0
cstheory.meta.stackexchange.com/._Users.xml:0
cstheory.meta.stackexchange.com/Votes.xml:0
cstheory.meta.stackexchange.com/Posts.xml:70
cstheory.meta.stackexchange.com/._Comments.xml:0
cstheory.meta.stackexchange.com/Badges.xml:0
cstheory.meta.stackexchange.com/Users.xml:73
cstheory.meta.stackexchange.com/Tags.xml:2
cstheory.meta.stackexchange.com/._PostHistory.xml:0
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 8

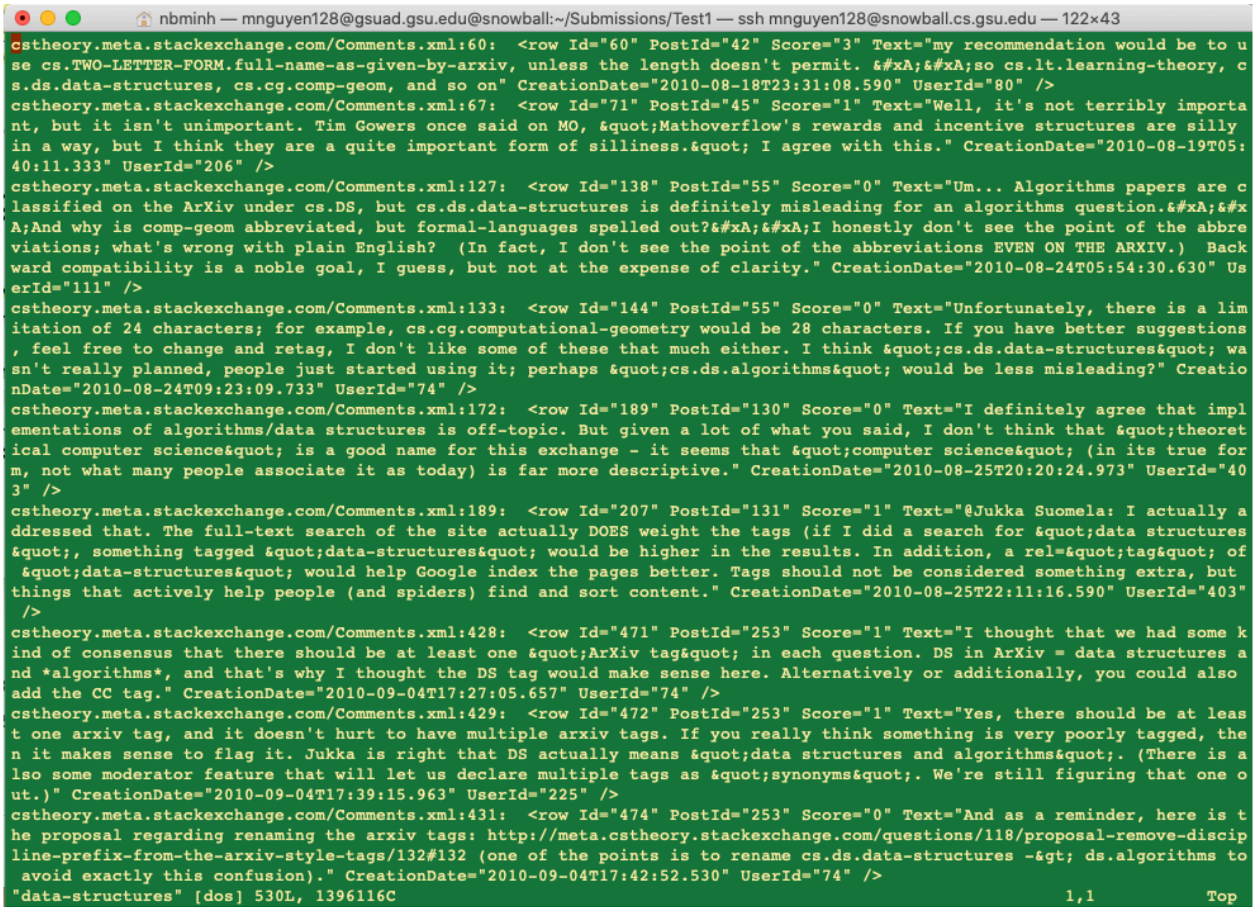
8. I then ran the grep command with “>” to parse the output to a file named “data-structures” (Figure 9)


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 122×43

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
computer-science  cstheory.meta.stackexchange.com  data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ grep -rnE "data|structures" cstheory.meta.stackexchange.com/ > data-structures
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
computer-science  cstheory.meta.stackexchange.com  data-structures  data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 9

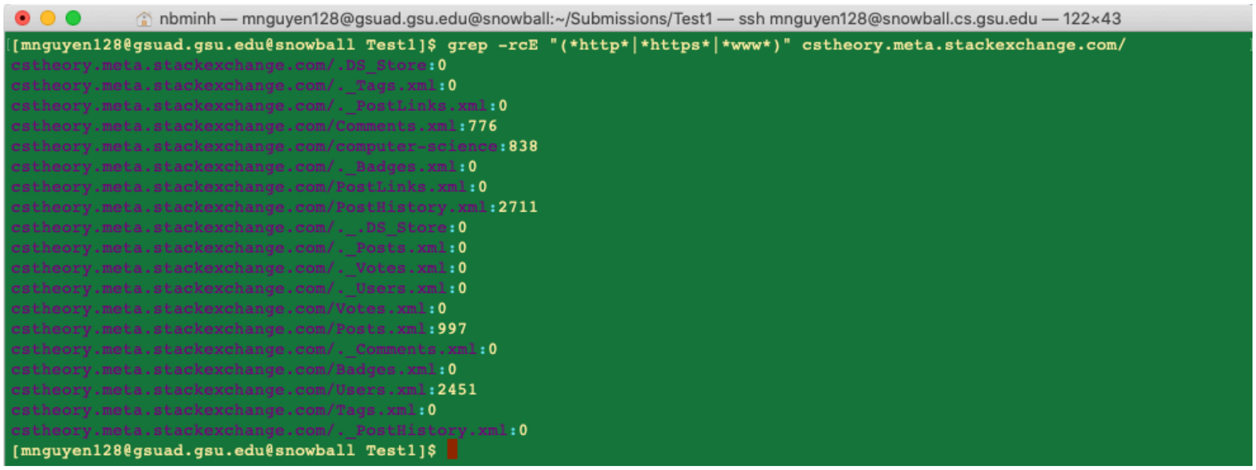
This is a quick look at the “data-structures” file (Figure 10).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 122×43

```
cstheory.meta.stackexchange.com/Comments.xml:60:  <row Id="60" PostId="42" Score="3" Text="my recommendation would be to u
se cs.TWO-LETTER-FORM.full-name-as-given-by-arxiv, unless the length doesn't permit. &#xA;&#xA;so cs.lt.learning-theory, c
s.ds.data-structures, cs.cg.comp-geom, and so on" CreationDate="2010-08-18T23:31:08.590" UserId="80" />
cstheory.meta.stackexchange.com/Comments.xml:67:  <row Id="71" PostId="45" Score="1" Text="Well, it's not terribly importa
nt, but it isn't unimportant. Tim Gowers once said on MO, &quot;Mathoverflow's rewards and incentive structures are silly
in a way, but I think they are a quite important form of silliness.&quot; I agree with this." CreationDate="2010-08-19T05:
40:11.333" UserId="206" />
cstheory.meta.stackexchange.com/Comments.xml:127:  <row Id="138" PostId="55" Score="0" Text="Um... Algorithms papers are c
lassified on the ArXiv under cs.DS, but cs.ds.data-structures is definitely misleading for an algorithms question.&#xA;&#x
A;And why is comp-geom abbreviated, but formal-languages spelled out?&#xA;&#xA;I honestly don't see the point of the abbre
viations; what's wrong with plain English?  (In fact, I don't see the point of the abbreviations EVEN ON THE ARXIV.)  Back
ward compatibility is a noble goal, I guess, but not at the expense of clarity." CreationDate="2010-08-24T05:54:30.630" Us
erId="111" />
cstheory.meta.stackexchange.com/Comments.xml:133:  <row Id="144" PostId="55" Score="0" Text="Unfortunately, there is a lim
itation of 24 characters; for example, cs.cg.computational-geometry would be 28 characters. If you have better suggestions
, feel free to change and retag, I don't like some of these that much either. I think &quot;cs.ds.data-structures&quot; wa
sn't really planned, people just started using it; perhaps &quot;cs.ds.algorithms&quot; would be less misleading?" Creatio
nDate="2010-08-24T09:23:09.733" UserId="74" />
cstheory.meta.stackexchange.com/Comments.xml:172:  <row Id="189" PostId="130" Score="0" Text="I definitely agree that impl
ementations of algorithms/data structures is off-topic. But given a lot of what you said, I don't think that &quot;theoret
ical computer science&quot; is a good name for this exchange - it seems that &quot;computer science&quot; (in its true for
m, not what many people associate it as today) is far more descriptive." CreationDate="2010-08-25T20:20:24.973" UserId="40
3" />
cstheory.meta.stackexchange.com/Comments.xml:189:  <row Id="207" PostId="131" Score="1" Text="@Jukka Suomela: I actually a
ddressed that. The full-text search of the site actually DOES weight the tags (if I did a search for &quot;data structures
&quot;, something tagged &quot;data-structures&quot; would be higher in the results. In addition, a rel=&quot;tag&quot; of
 &quot;data-structures&quot; would help Google index the pages better. Tags should not be considered something extra, but
things that actively help people (and spiders) find and sort content." CreationDate="2010-08-25T22:11:16.590" UserId="403"
 />
cstheory.meta.stackexchange.com/Comments.xml:428:  <row Id="471" PostId="253" Score="1" Text="I thought that we had some k
ind of consensus that there should be at least one &quot;ArXiv tag&quot; in each question. DS in ArXiv = data structures a
nd *algorithms*, and that's why I thought the DS tag would make sense here. Alternatively or additionally, you could also
add the CC tag." CreationDate="2010-09-04T17:27:05.657" UserId="74" />
cstheory.meta.stackexchange.com/Comments.xml:429:  <row Id="472" PostId="253" Score="1" Text="Yes, there should be at leas
t one arxiv tag, and it doesn't hurt to have multiple arxiv tags. If you really think something is very poorly tagged, the
n it makes sense to flag it. Jukka is right that DS actually means &quot;data structures and algorithms&quot;. (There is a
lso some moderator feature that will let us declare multiple tags as &quot;synonyms&quot;. We're still figuring that one o
ut.)" CreationDate="2010-09-04T17:39:15.963" UserId="225" />
cstheory.meta.stackexchange.com/Comments.xml:431:  <row Id="474" PostId="253" Score="0" Text="And as a reminder, here is t
he proposal regarding renaming the arxiv tags: http://meta.cstheory.stackexchange.com/questions/118/proposal-remove-discip
line-prefix-from-the-arxiv-style-tags/132#132 (one of the points is to rename cs.ds.data-structures -&gt; ds.algorithms to
 avoid exactly this confusion)." CreationDate="2010-09-04T17:42:52.530" UserId="74" />
"data-structures" [dos] 530L, 1396116C                                                              1,1           Top
```

Figure 10

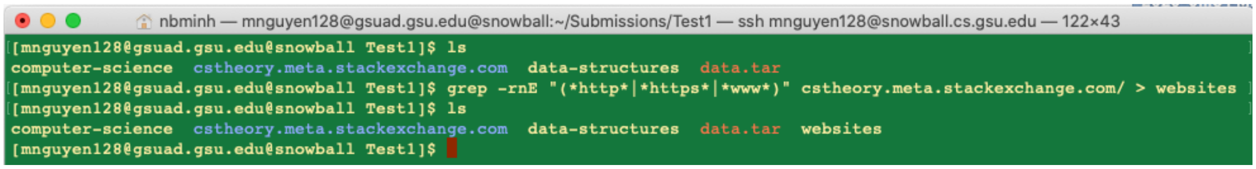
9. I then ran a grep command with a regular pattern to count how many lines contain links to websites (Figure 11).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 122×43

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ grep -rcE "(*http*|*https*|*www*)" cstheory.meta.stackexchange.com/
cstheory.meta.stackexchange.com/.DS_Store:0
cstheory.meta.stackexchange.com/._Tags.xml:0
cstheory.meta.stackexchange.com/._PostLinks.xml:0
cstheory.meta.stackexchange.com/Comments.xml:776
cstheory.meta.stackexchange.com/computer-science:838
cstheory.meta.stackexchange.com/._Badges.xml:0
cstheory.meta.stackexchange.com/PostLinks.xml:0
cstheory.meta.stackexchange.com/PostHistory.xml:2711
cstheory.meta.stackexchange.com/._.DS_Store:0
cstheory.meta.stackexchange.com/._Posts.xml:0
cstheory.meta.stackexchange.com/._Votes.xml:0
cstheory.meta.stackexchange.com/._Users.xml:0
cstheory.meta.stackexchange.com/Votes.xml:0
cstheory.meta.stackexchange.com/Posts.xml:997
cstheory.meta.stackexchange.com/._Comments.xml:0
cstheory.meta.stackexchange.com/Badges.xml:0
cstheory.meta.stackexchange.com/Users.xml:2451
cstheory.meta.stackexchange.com/Tags.xml:0
cstheory.meta.stackexchange.com/._PostHistory.xml:0
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 11

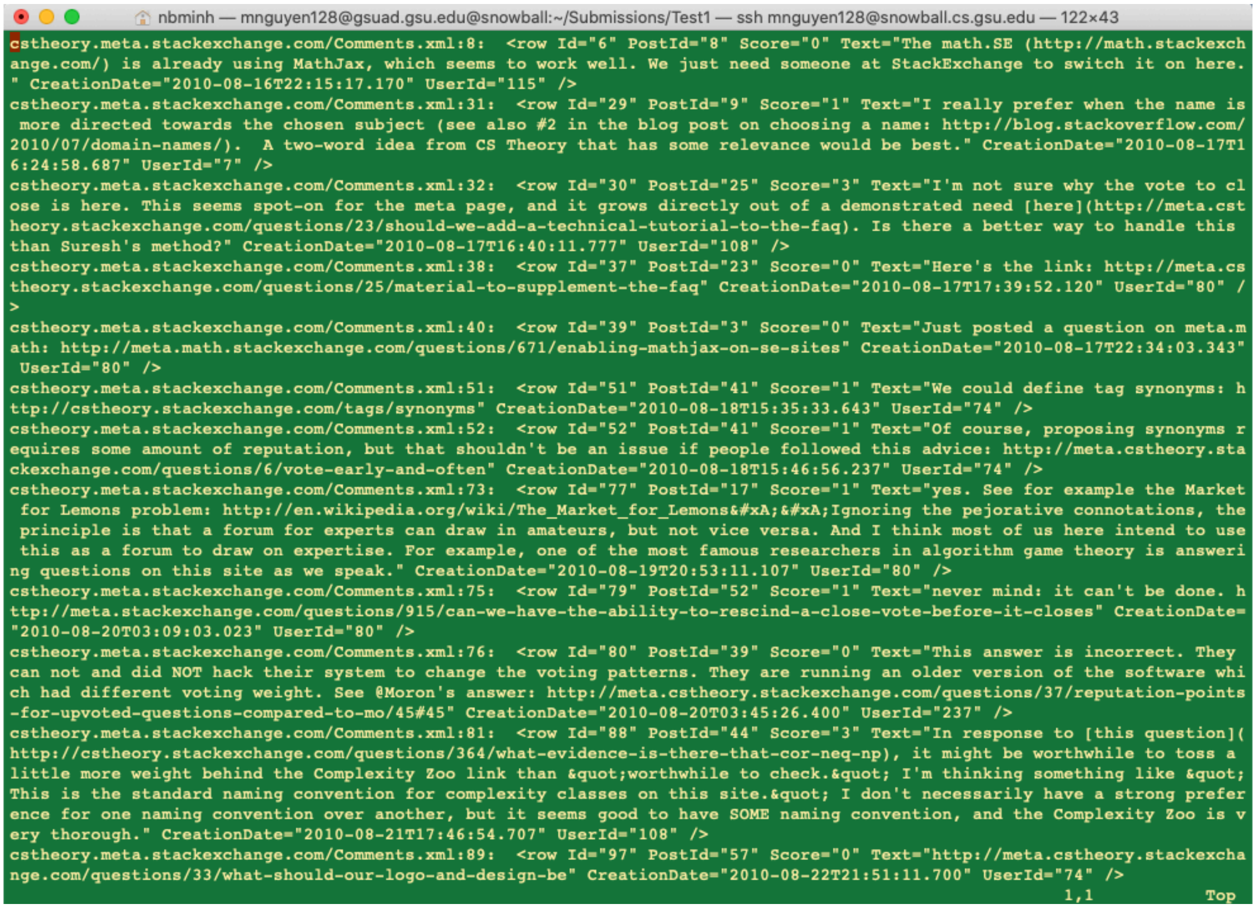
10. I then ran the grep command with “>” to parse the output to a file named “websites” (Figure 12).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 122×43

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
computer-science  cstheory.meta.stackexchange.com  data-structures  data.tar
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ grep -rnE "(*http*|*https*|*www*)" cstheory.meta.stackexchange.com/ > websites
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
computer-science  cstheory.meta.stackexchange.com  data-structures  data.tar  websites
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ 
```

Figure 12

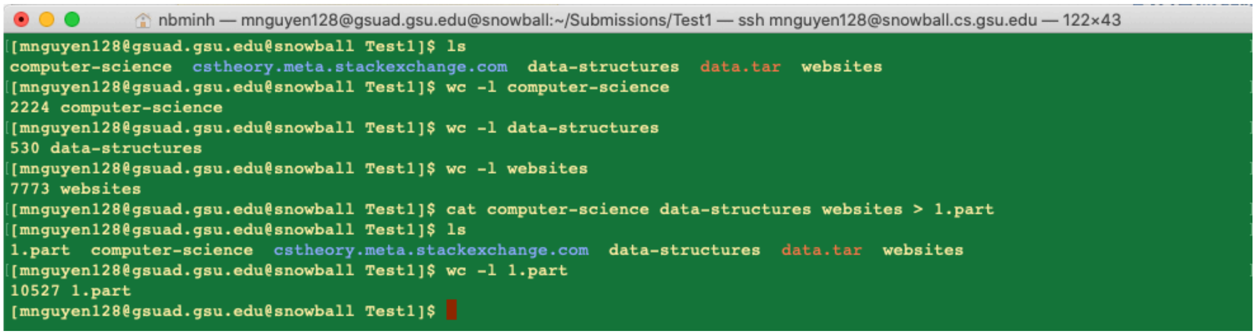
This is a quick look at the “websites” file (Figure 13).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 122×43

```
cstheory.meta.stackexchange.com/Comments.xml:8:  <row Id="6" PostId="8" Score="0" Text="The math.SE (http://math.stackexch
ange.com/) is already using MathJax, which seems to work well. We just need someone at StackExchange to switch it on here.
" CreationDate="2010-08-16T22:15:17.170" UserId="115" />
cstheory.meta.stackexchange.com/Comments.xml:31:  <row Id="29" PostId="9" Score="1" Text="I really prefer when the name is
 more directed towards the chosen subject (see also #2 in the blog post on choosing a name: http://blog.stackoverflow.com/
2010/07/domain-names/).  A two-word idea from CS Theory that has some relevance would be best." CreationDate="2010-08-17T1
6:24:58.687" UserId="7" />
cstheory.meta.stackexchange.com/Comments.xml:32:  <row Id="30" PostId="25" Score="3" Text="I'm not sure why the vote to cl
ose is here. This seems spot-on for the meta page, and it grows directly out of a demonstrated need [here](http://meta.cst
heory.stackexchange.com/questions/23/should-we-add-a-technical-tutorial-to-the-faq). Is there a better way to handle this
than Suresh's method?" CreationDate="2010-08-17T16:40:11.777" UserId="108" />
cstheory.meta.stackexchange.com/Comments.xml:38:  <row Id="37" PostId="23" Score="0" Text="Here's the link: http://meta.cs
theory.stackexchange.com/questions/25/material-to-supplement-the-faq" CreationDate="2010-08-17T17:39:52.120" UserId="80" /
>
cstheory.meta.stackexchange.com/Comments.xml:40:  <row Id="39" PostId="3" Score="0" Text="Just posted a question on meta.m
ath: http://meta.math.stackexchange.com/questions/671/enabling-mathjax-on-se-sites" CreationDate="2010-08-17T22:34:03.343"
 UserId="80" />
cstheory.meta.stackexchange.com/Comments.xml:51:  <row Id="51" PostId="41" Score="1" Text="We could define tag synonyms: h
ttp://cstheory.stackexchange.com/tags/synonyms" CreationDate="2010-08-18T15:35:33.643" UserId="74" />
cstheory.meta.stackexchange.com/Comments.xml:52:  <row Id="52" PostId="41" Score="1" Text="Of course, proposing synonyms r
equires some amount of reputation, but that shouldn't be an issue if people followed this advice: http://meta.cstheory.sta
ckexchange.com/questions/6/vote-early-and-often" CreationDate="2010-08-18T15:46:56.237" UserId="74" />
cstheory.meta.stackexchange.com/Comments.xml:73:  <row Id="77" PostId="17" Score="1" Text="yes. See for example the Market
 for Lemons problem: http://en.wikipedia.org/wiki/The_Market_for_Lemons&#xA;&#xA;Ignoring the pejorative connotations, the
 principle is that a forum for experts can draw in amateurs, but not vice versa. And I think most of us here intend to use
 this as a forum to draw on expertise. For example, one of the most famous researchers in algorithm game theory is answeri
ng questions on this site as we speak." CreationDate="2010-08-19T20:53:11.107" UserId="80" />
cstheory.meta.stackexchange.com/Comments.xml:75:  <row Id="79" PostId="52" Score="1" Text="never mind: it can't be done. h
ttp://meta.stackexchange.com/questions/915/can-we-have-the-ability-to-rescind-a-close-vote-before-it-closes" CreationDate=
"2010-08-20T03:09:03.023" UserId="80" />
cstheory.meta.stackexchange.com/Comments.xml:76:  <row Id="80" PostId="39" Score="0" Text="This answer is incorrect. They
can not and did NOT hack their system to change the voting patterns. They are running an older version of the software whi
ch had different voting weight. See @Moron's answer: http://meta.cstheory.stackexchange.com/questions/37/reputation-points
-for-upvoted-questions-compared-to-mo/45#45" CreationDate="2010-08-20T03:45:26.400" UserId="237" />
cstheory.meta.stackexchange.com/Comments.xml:81:  <row Id="88" PostId="44" Score="3" Text="In response to [this question](
http://cstheory.stackexchange.com/questions/364/what-evidence-is-there-that-cor-neq-np), it might be worthwhile to toss a
little more weight behind the Complexity Zoo link than &quot;worthwhile to check.&quot; I'm thinking something like &quot;
This is the standard naming convention for complexity classes on this site.&quot; I don't necessarily have a strong prefer
ence for one naming convention over another, but it seems good to have SOME naming convention, and the Complexity Zoo is v
ery thorough." CreationDate="2010-08-21T17:46:54.707" UserId="108" />
cstheory.meta.stackexchange.com/Comments.xml:89:  <row Id="97" PostId="57" Score="0" Text="http://meta.cstheory.stackexcha
nge.com/questions/33/what-should-our-logo-and-design-be" CreationDate="2010-08-22T21:51:11.700" UserId="74" />
                                                                                                1,1           Top
```

Figure 13

11. Finally, I combined all the results from “websites”, “data-structures”, and “computer-science” in a file named “1.part” using “cat” command. As you can see, the “1.part” file has the sum of all the lines that the other 3 files have (Figure 14).

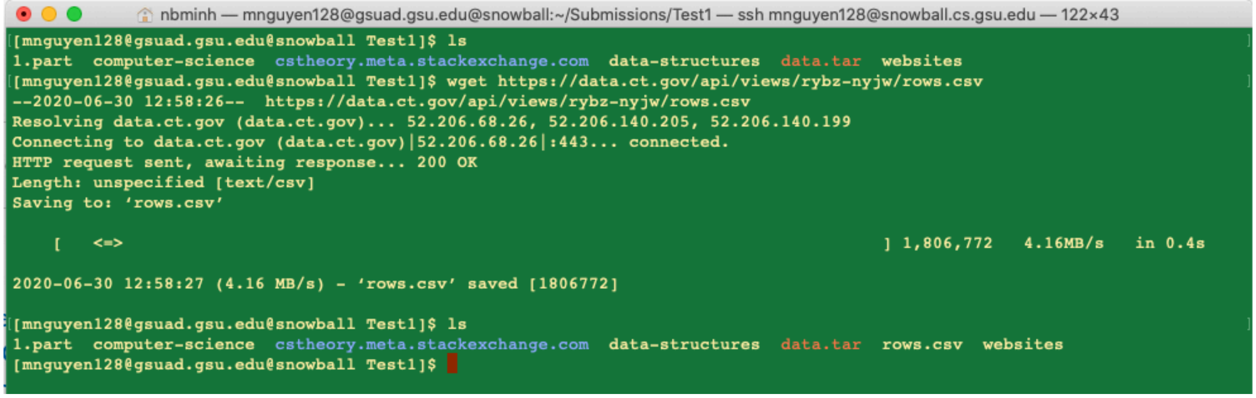

nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 122×43

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
computer-science  cstheory.meta.stackexchange.com  data-structures  data.tar  websites
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ wc -l computer-science
2224 computer-science
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ wc -l data-structures
530 data-structures
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ wc -l websites
7773 websites
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ cat computer-science data-structures websites > 1.part
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
1.part  computer-science  cstheory.meta.stackexchange.com  data-structures  data.tar  websites
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ wc -l 1.part
10527 1.part
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 14

Part 2

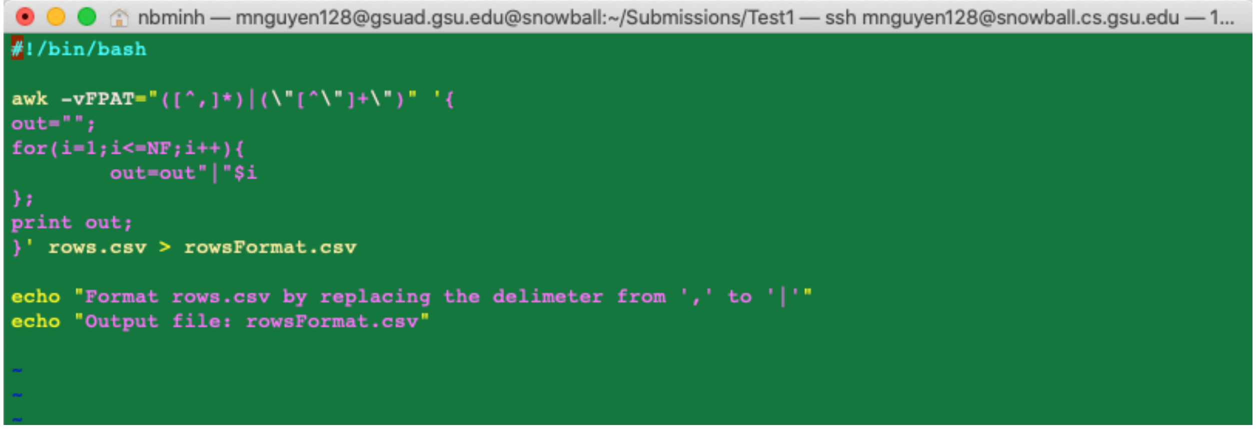
1. I used the “wget” command to download the “rows.csv” file from the URL (Figure 15).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 122×43

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
1.part  computer-science  cstheory.meta.stackexchange.com  data-structures  data.tar  websites
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ wget https://data.ct.gov/api/views/rybz-nyjw/rows.csv
--2020-06-30 12:58:26--  https://data.ct.gov/api/views/rybz-nyjw/rows.csv
Resolving data.ct.gov (data.ct.gov)... 52.206.68.26, 52.206.140.205, 52.206.140.199
Connecting to data.ct.gov (data.ct.gov)|52.206.68.26|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/csv]
Saving to: ‘rows.csv’

    [   <=>                                                                            ] 1,806,772   4.16MB/s   in 0.4s

2020-06-30 12:58:27 (4.16 MB/s) - ‘rows.csv’ saved [1806772]

[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
1.part  computer-science  cstheory.meta.stackexchange.com  data-structures  data.tar  rows.csv  websites
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 15

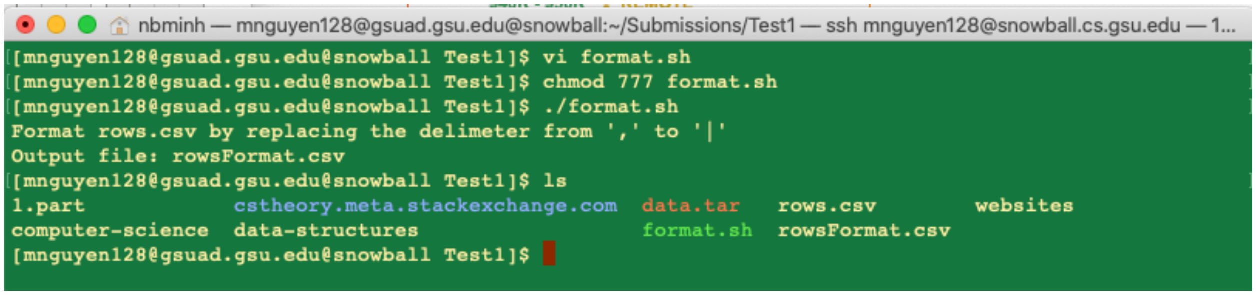
2. I then wrote a script to format the “rows.csv” file by replacing all the commas that are used to separate the columns to the “|” symbol in order to make the file easy to read. Then I saved the output to a file named “rowsFormat.csv” for later uses. This is the “format.sh” script that I wrote using awk (Figure 16).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 1...

```
#!/bin/bash

awk -vFPAT="([^,]*)|(\"[^\"]+\")" '{
out="";
for(i=1;i<=NF;i++){
        out=out"|"$i
};
print out;
}' rows.csv > rowsFormat.csv

echo "Format rows.csv by replacing the delimeter from ',' to '|'"
echo "Output file: rowsFormat.csv"

~
~
~
```

Figure 16

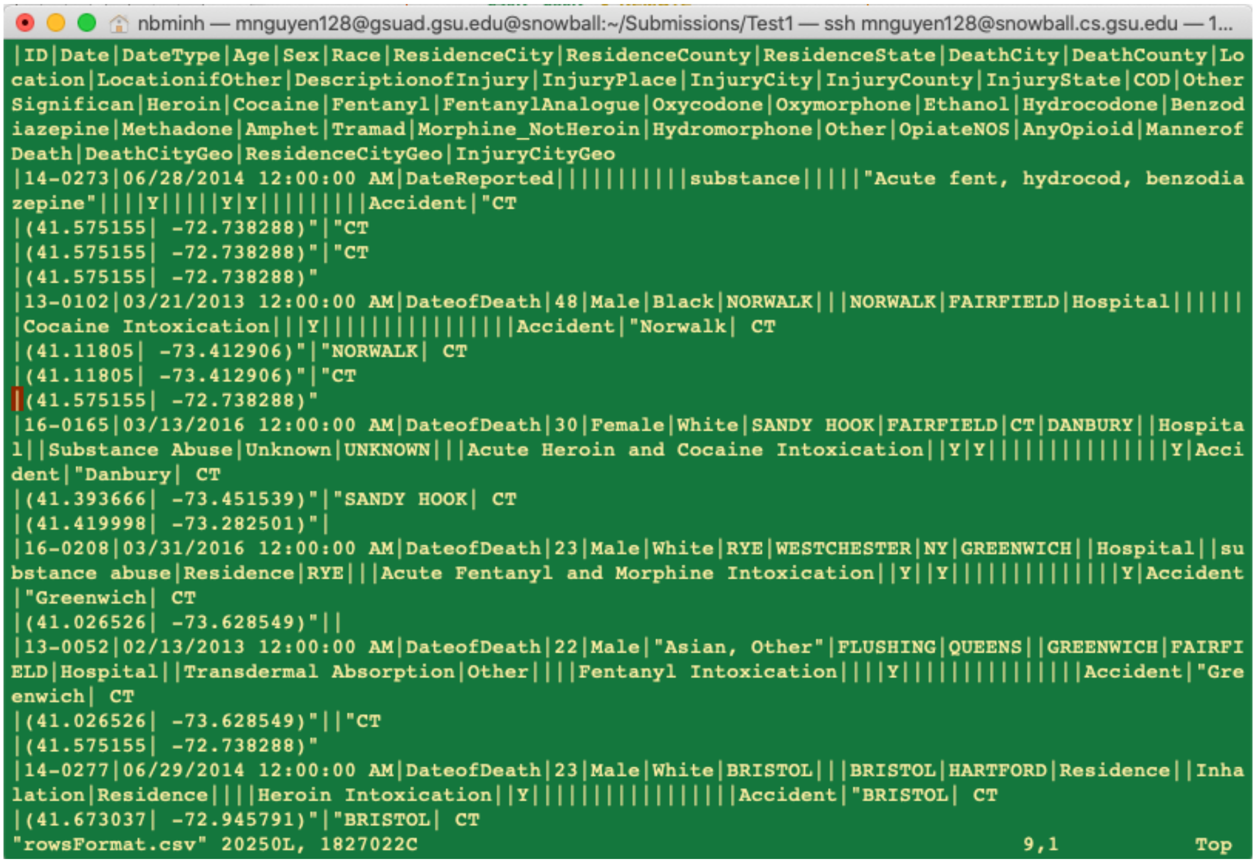
And this is the actual run of the script. Note that I changed the permissions so that I can execute the script (Figure 17).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 1...

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ vi format.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ chmod 777 format.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ./format.sh
Format rows.csv by replacing the delimeter from ',' to '|'
Output file: rowsFormat.csv
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
1.part            cstheory.meta.stackexchange.com  data.tar   rows.csv        websites
computer-science  data-structures                  format.sh  rowsFormat.csv
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ 
```

Figure 17

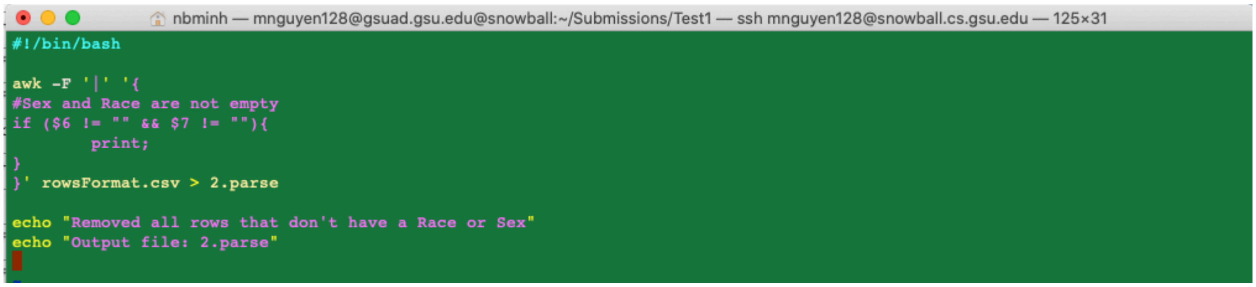
And this is a quick look at the “rowsFormat.csv” file (Figure 18).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 1...

```
|ID|Date|DateType|Age|Sex|Race|ResidenceCity|ResidenceCounty|ResidenceState|DeathCity|DeathCounty|Lo
cation|LocationifOther|DescriptionofInjury|InjuryPlace|InjuryCity|InjuryCounty|InjuryState|COD|Other
Significan|Heroin|Cocaine|Fentanyl|FentanylAnalogue|Oxycodone|Oxymorphone|Ethanol|Hydrocodone|Benzod
iazepine|Methadone|Amphet|Tramad|Morphine_NotHeroin|Hydromorphone|Other|OpiateNOS|AnyOpioid|Mannerof
Death|DeathCityGeo|ResidenceCityGeo|InjuryCityGeo
|14-0273|06/28/2014 12:00:00 AM|DateReported||||||||||substance||||"Acute fent, hydrocod, benzodia
zepine"||||Y|||||Y|Y||||||||Accident|"CT
|(41.575155| -72.738288)"|"CT
|(41.575155| -72.738288)"|"CT
|(41.575155| -72.738288)"
|13-0102|03/21/2013 12:00:00 AM|DateofDeath|48|Male|Black|NORWALK||NORWALK|FAIRFIELD|Hospital|||||
|Cocaine Intoxication||Y||||||||||||||Accident|"Norwalk| CT
|(41.11805| -73.412906)"|"NORWALK| CT
|(41.11805| -73.412906)"|"CT
|(41.575155| -72.738288)"
|16-0165|03/13/2016 12:00:00 AM|DateofDeath|30|Female|White|SANDY HOOK|FAIRFIELD|CT|DANBURY||Hospita
l||Substance Abuse|Unknown|UNKNOWN||Acute Heroin and Cocaine Intoxication||Y|Y|||||||||||||Y|Acci
dent|"Danbury| CT
|(41.393666| -73.451539)"|"SANDY HOOK| CT
|(41.419998| -73.282501)"|
|16-0208|03/31/2016 12:00:00 AM|DateofDeath|23|Male|White|RYE|WESTCHESTER|NY|GREENWICH||Hospital||su
bstance abuse|Residence|RYE||Acute Fentanyl and Morphine Intoxication||Y||Y|||||||||||||Y|Accident
|"Greenwich| CT
|(41.026526| -73.628549)"||
|13-0052|02/13/2013 12:00:00 AM|DateofDeath|22|Male|"Asian, Other"|FLUSHING|QUEENS||GREENWICH|FAIRFI
ELD|Hospital||Transdermal Absorption|Other|||Fentanyl Intoxication|||Y|||||||||||||Accident|"Gre
enwich| CT
|(41.026526| -73.628549)"||"CT
|(41.575155| -72.738288)"
|14-0277|06/29/2014 12:00:00 AM|DateofDeath|23|Male|White|BRISTOL||BRISTOL|HARTFORD|Residence||Inha
lation|Residence|||Heroin Intoxication||Y||||||||||||||Accident|"BRISTOL| CT
|(41.673037| -72.945791)"|"BRISTOL| CT
"rowsFormat.csv" 20250L, 1827022C                                               9,1            Top
```

Figure 18

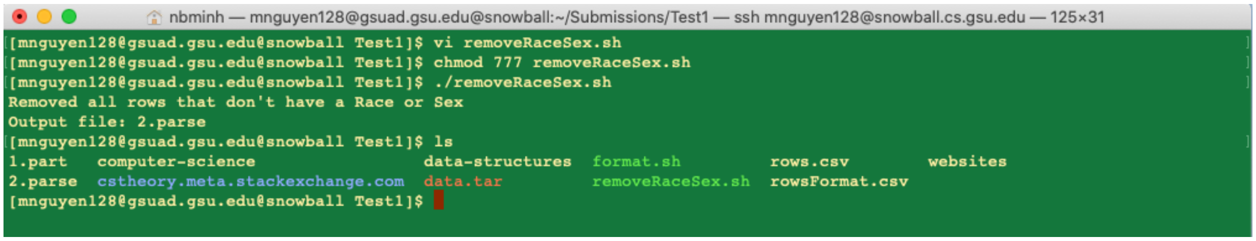
3. I then wrote a script to remove all the lines that don’t have a Race or Sex named “removeRaceSex.sh”. The script will parse the output to a file named “2.parse” as required (Figure 19).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
#!/bin/bash

awk -F '|' '{
#Sex and Race are not empty
if ($6 != "" && $7 != ""){
        print;
}
}' rowsFormat.csv > 2.parse

echo "Removed all rows that don't have a Race or Sex"
echo "Output file: 2.parse"
```

Figure 19

And this is the actual run of the script. Note that I changed the permissions so that I can execute the script (Figure 20).

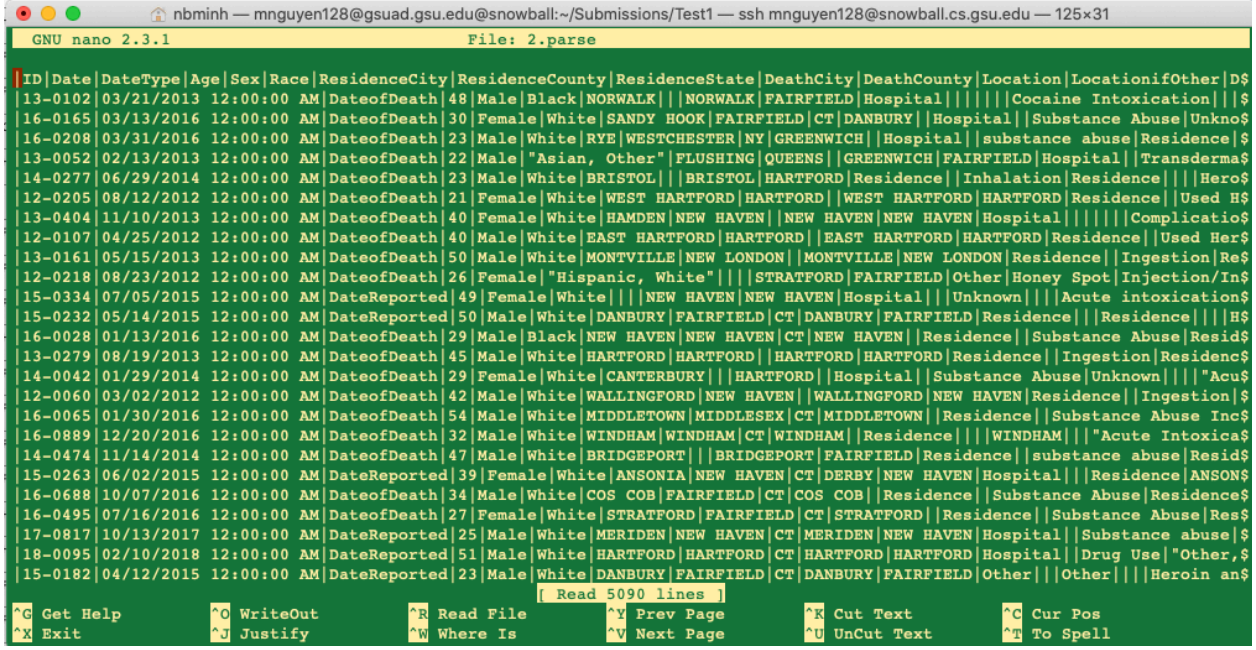


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ vi removeRaceSex.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ chmod 777 removeRaceSex.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ./removeRaceSex.sh
Removed all rows that don't have a Race or Sex
Output file: 2.parse
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
1.part   computer-science                 data-structures  format.sh         rows.csv        websites
2.parse  cstheory.meta.stackexchange.com  data.tar         removeRaceSex.sh  rowsFormat.csv
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 20

And this is a quick look at the “2.parse” file (Figure 21).

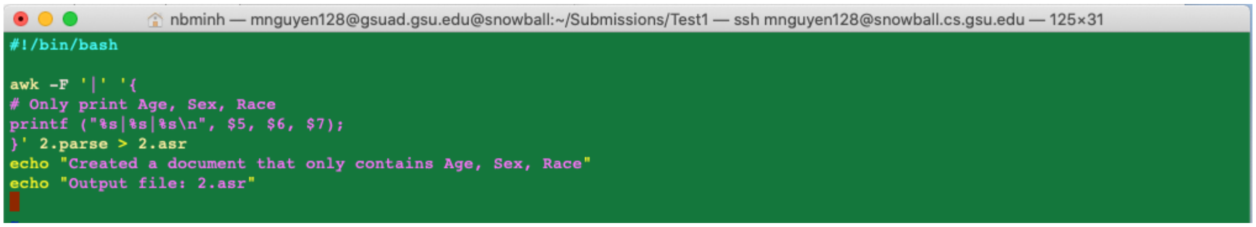


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
  GNU nano 2.3.1                              File: 2.parse

|ID|Date|DateType|Age|Sex|Race|ResidenceCity|ResidenceCounty|ResidenceState|DeathCity|DeathCounty|Location|LocationifOther|D$
|13-0102|03/21/2013 12:00:00 AM|DateofDeath|48|Male|Black|NORWALK|||NORWALK|FAIRFIELD|Hospital||||||Cocaine Intoxication|||$
|16-0165|03/13/2016 12:00:00 AM|DateofDeath|30|Female|White|SANDY HOOK|FAIRFIELD|CT|DANBURY||Hospital||Substance Abuse|Unkno$
|16-0208|03/31/2016 12:00:00 AM|DateofDeath|23|Male|White|RYE|WESTCHESTER|NY|GREENWICH||Hospital||substance abuse|Residence|$
|13-0052|02/13/2013 12:00:00 AM|DateofDeath|22|Male|"Asian, Other"|FLUSHING|QUEENS||GREENWICH|FAIRFIELD|Hospital||Transderma$
|14-0277|06/29/2014 12:00:00 AM|DateofDeath|23|Male|White|BRISTOL|||BRISTOL|HARTFORD|Residence||Inhalation|Residence|||Hero$
|12-0205|08/12/2012 12:00:00 AM|DateofDeath|21|Female|White|WEST HARTFORD|HARTFORD||WEST HARTFORD|HARTFORD|Residence||Used H$
|13-0404|11/10/2013 12:00:00 AM|DateofDeath|40|Female|White|HAMDEN|NEW HAVEN||NEW HAVEN|NEW HAVEN|Hospital||||||Complicatio$
|12-0107|04/25/2012 12:00:00 AM|DateofDeath|40|Male|White|EAST HARTFORD|HARTFORD||EAST HARTFORD|HARTFORD|Residence||Used Her$
|13-0161|05/15/2013 12:00:00 AM|DateofDeath|50|Male|White|MONTVILLE|NEW LONDON||MONTVILLE|NEW LONDON|Residence||Ingestion|Re$
|12-0218|08/23/2012 12:00:00 AM|DateofDeath|26|Female|"Hispanic, White"||||STRATFORD|FAIRFIELD|Other|Honey Spot|Injection/In$
|15-0334|07/05/2015 12:00:00 AM|DateReported|49|Female|White|||NEW HAVEN|NEW HAVEN|Hospital||Unknown|||Acute intoxication$
|15-0232|05/14/2015 12:00:00 AM|DateReported|50|Male|White|DANBURY|FAIRFIELD|CT|DANBURY|FAIRFIELD|Residence||Residence|||H$
|16-0028|01/13/2016 12:00:00 AM|DateofDeath|29|Male|Black|NEW HAVEN|NEW HAVEN|CT|NEW HAVEN||Residence||Substance Abuse|Resid$
|13-0279|08/19/2013 12:00:00 AM|DateofDeath|45|Male|White|HARTFORD|HARTFORD||HARTFORD|HARTFORD|Residence||Ingestion|Residenc$
|14-0042|01/29/2014 12:00:00 AM|DateofDeath|29|Female|White|CANTERBURY|||HARTFORD||Hospital||Substance Abuse|Unknown||||"Acu$
|12-0060|03/02/2012 12:00:00 AM|DateofDeath|42|Male|White|WALLINGFORD|NEW HAVEN||WALLINGFORD|NEW HAVEN|Residence||Ingestion|$
|16-0065|01/30/2016 12:00:00 AM|DateofDeath|54|Male|White|MIDDLETOWN|MIDDLESEX|CT|MIDDLETOWN||Residence||Substance Abuse Inc$
|16-0889|12/20/2016 12:00:00 AM|DateofDeath|32|Male|White|WINDHAM|WINDHAM|CT|WINDHAM||Residence|||WINDHAM||"Acute Intoxica$
|14-0474|11/14/2014 12:00:00 AM|DateofDeath|47|Male|White|BRIDGEPORT|||BRIDGEPORT|FAIRFIELD|Residence||substance abuse|Resid$
|15-0263|06/02/2015 12:00:00 AM|DateReported|39|Female|White|ANSONIA|NEW HAVEN|CT|DERBY|NEW HAVEN|Hospital||Residence|ANSON$
|16-0688|10/07/2016 12:00:00 AM|DateofDeath|34|Male|White|COS COB|FAIRFIELD|CT|COS COB||Residence||Substance Abuse|Residence$
|16-0495|07/16/2016 12:00:00 AM|DateofDeath|27|Female|White|STRATFORD|FAIRFIELD|CT|STRATFORD||Residence||Substance Abuse|Res$
|17-0817|10/13/2017 12:00:00 AM|DateReported|25|Male|White|MERIDEN|NEW HAVEN|CT|MERIDEN|NEW HAVEN|Hospital||Substance abuse|$
|18-0095|02/10/2018 12:00:00 AM|DateReported|51|Male|White|HARTFORD|HARTFORD|CT|HARTFORD|HARTFORD|Hospital||Drug Use|"Other,$
|15-0182|04/12/2015 12:00:00 AM|DateReported|23|Male|White|DANBURY|FAIRFIELD|CT|DANBURY|FAIRFIELD|Other||Other|||Heroin an$
                                                  [ Read 5090 lines ]
^G Get Help         ^O WriteOut         ^R Read File        ^Y Prev Page        ^K Cut Text         ^C Cur Pos
^X Exit             ^J Justify          ^W Where Is         ^V Next Page        ^U UnCut Text       ^T To Spell
```

Figure 21

4. I then wrote a script to create a new document that only contains age, sex, race, then save the output to a file named “2.asr”. This is my “createDoc.sh” script (Figure 22).

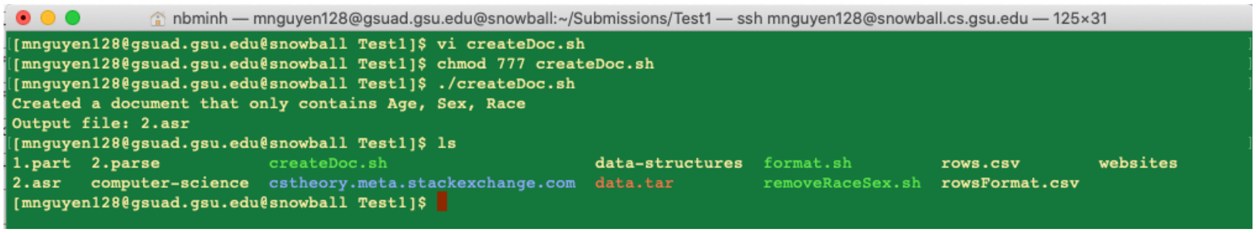


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
#!/bin/bash

awk -F '|' '{
# Only print Age, Sex, Race
printf ("%s|%s|%s\n", $5, $6, $7);
}' 2.parse > 2.asr
echo "Created a document that only contains Age, Sex, Race"
echo "Output file: 2.asr"
```

Figure 22

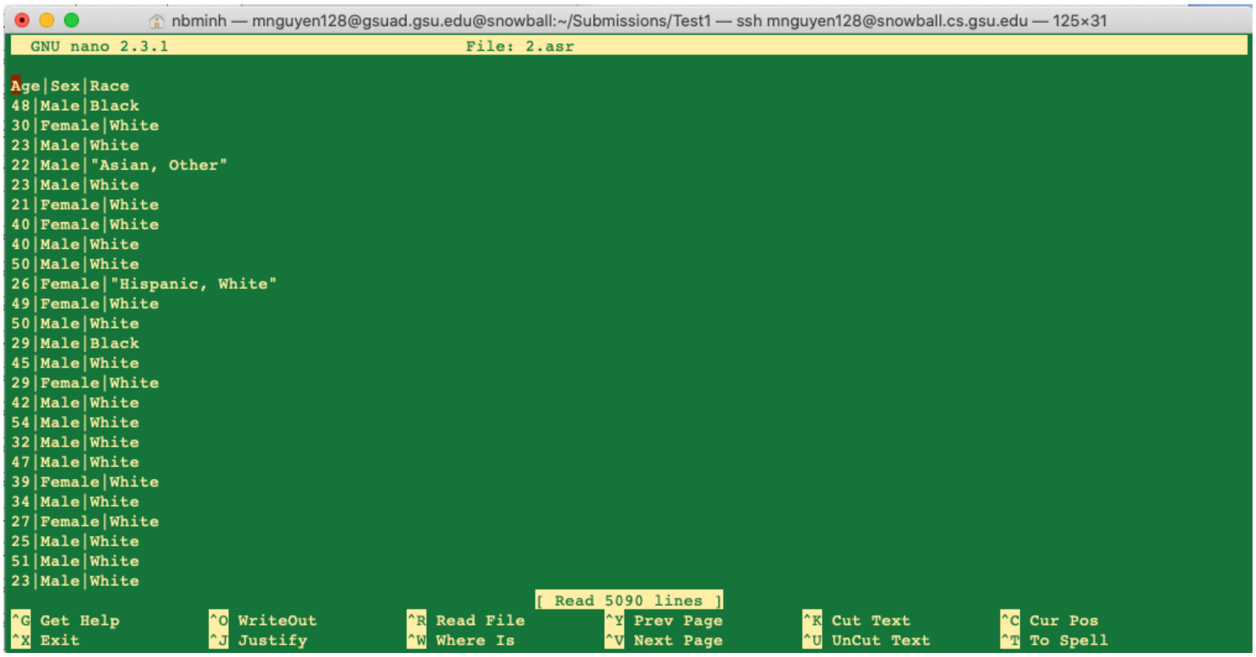
And this is the actual run of the script. Note that I changed the permissions so that I can execute the script (Figure 23).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ vi createDoc.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ chmod 777 createDoc.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ./createDoc.sh
Created a document that only contains Age, Sex, Race
Output file: 2.asr
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
1.part  2.parse           createDoc.sh                     data-structures  format.sh         rows.csv        websites
2.asr   computer-science  cstheory.meta.stackexchange.com  data.tar         removeRaceSex.sh  rowsFormat.csv
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ 
```

Figure 23

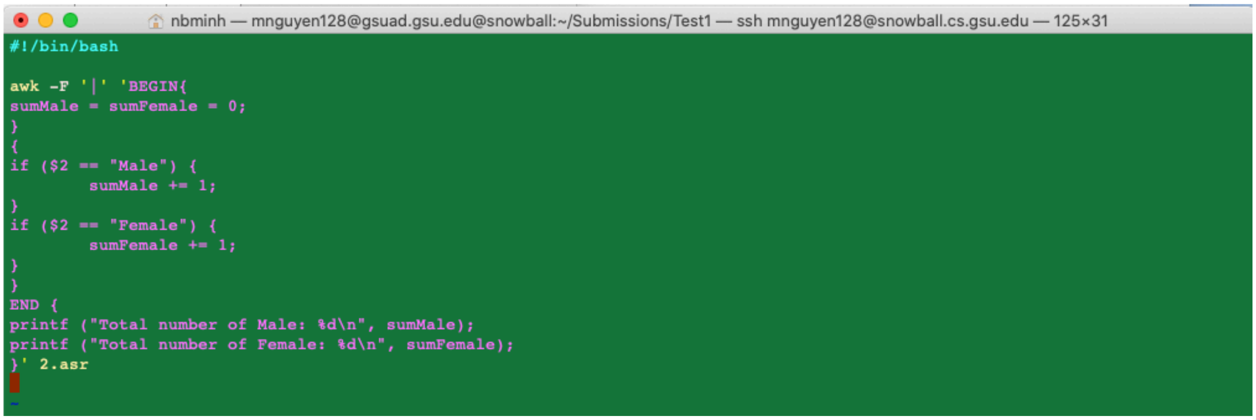
And this is a quick look at the “2.asr” file (Figure 24).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
  GNU nano 2.3.1                            File: 2.asr

Age|Sex|Race
48|Male|Black
30|Female|White
23|Male|White
22|Male|"Asian, Other"
23|Male|White
21|Female|White
40|Female|White
40|Male|White
50|Male|White
26|Female|"Hispanic, White"
49|Female|White
50|Male|White
29|Male|Black
45|Male|White
29|Female|White
42|Male|White
54|Male|White
32|Male|White
47|Male|White
39|Female|White
34|Male|White
27|Female|White
25|Male|White
51|Male|White
23|Male|White
                                                    [ Read 5090 lines ]
^G Get Help         ^O WriteOut         ^R Read File        ^Y Prev Page        ^K Cut Text         ^C Cur Pos
^X Exit             ^J Justify          ^W Where Is         ^V Next Page        ^U UnCut Text       ^T To Spell
```

Figure 24

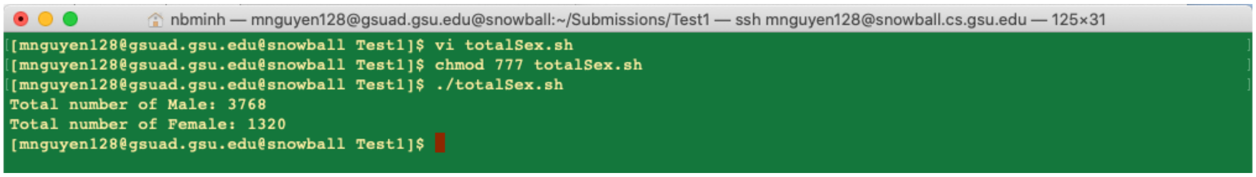
5. I wrote a script to find the total number of a Male and Female participants, separately using awk scripting language, and it takes the “2.asr” as input file. This is my “totalSex.sh” script (Figure 25).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
#!/bin/bash

awk -F '|' 'BEGIN{
sumMale = sumFemale = 0;
}
{
if ($2 == "Male") {
        sumMale += 1;
}
if ($2 == "Female") {
        sumFemale += 1;
}
}
END {
printf ("Total number of Male: %d\n", sumMale);
printf ("Total number of Female: %d\n", sumFemale);
}' 2.asr

~
```

Figure 25

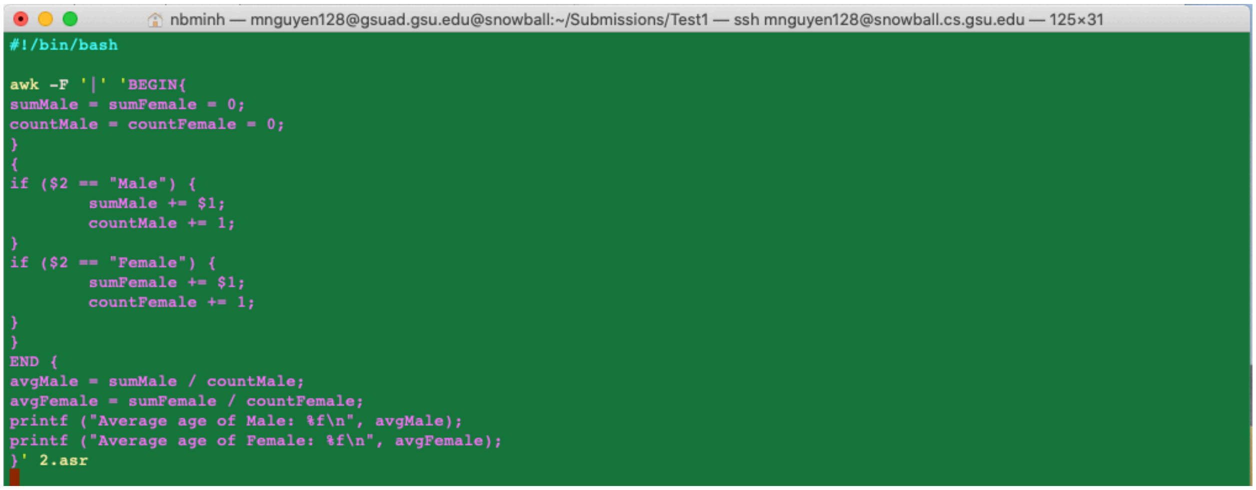
And this is the actual run of the script. Note that I changed the permissions so that I can execute the script (Figure 26).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ vi totalSex.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ chmod 777 totalSex.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ./totalSex.sh
Total number of Male: 3768
Total number of Female: 1320
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 26

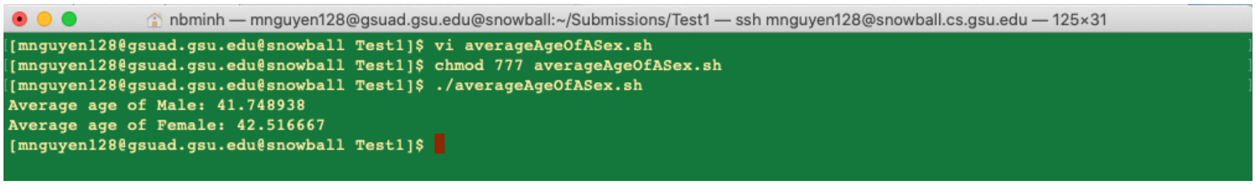
6. Then I wrote another script named “averageAgeOfASex.sh” to find the average age of a Male and Female participants, separately using awk (Figure 27).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
#!/bin/bash

awk -F '|' 'BEGIN{
sumMale = sumFemale = 0;
countMale = countFemale = 0;
}
{
if ($2 == "Male") {
        sumMale += $1;
        countMale += 1;
}
if ($2 == "Female") {
        sumFemale += $1;
        countFemale += 1;
}
}
END {
avgMale = sumMale / countMale;
avgFemale = sumFemale / countFemale;
printf ("Average age of Male: %f\n", avgMale);
printf ("Average age of Female: %f\n", avgFemale);
}' 2.asr
```

Figure 27

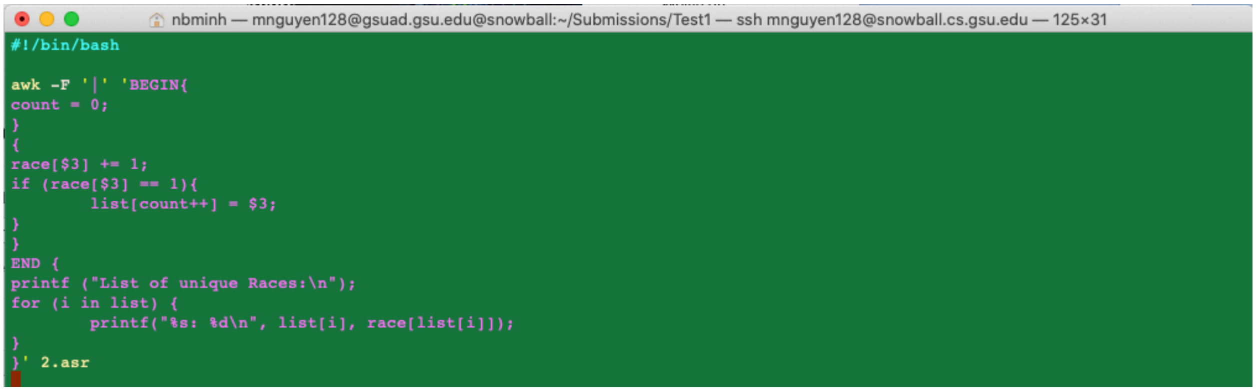
And this is the actual run of the script. Note that I changed the permissions so that I can execute the script (Figure 28).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ vi averageAgeOfASex.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ chmod 777 averageAgeOfASex.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ./averageAgeOfASex.sh
Average age of Male: 41.748938
Average age of Female: 42.516667
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 28

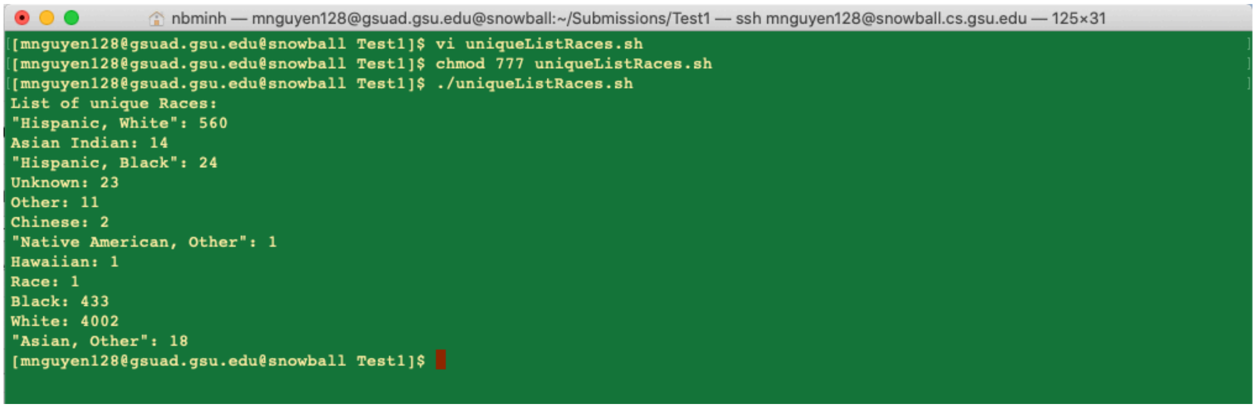
7. I wrote a script named “uniqueListRaces.sh” to find the list of the unique races that are in the study using awk and array (dictionary). Every race is a key of the dictionary and is counted to keep track of its existence (Figure 29).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
#!/bin/bash

awk -F '|' 'BEGIN{
count = 0;
}
{
race[$3] += 1;
if (race[$3] == 1){
        list[count++] = $3;
}
}
END {
printf ("List of unique Races:\n");
for (i in list) {
        printf("%s: %d\n", list[i], race[list[i]]);
}
}' 2.asr
```

Figure 29

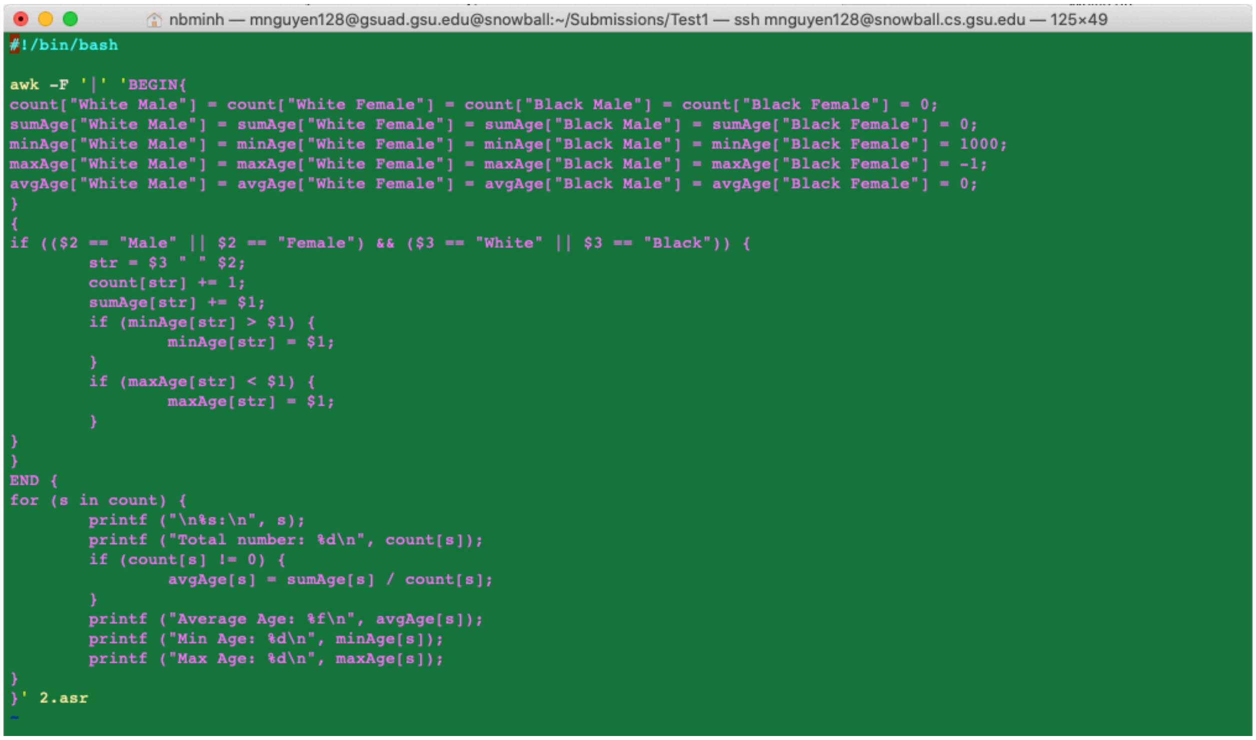
And this is the actual run of the script. Note that I changed the permissions so that I can execute the script (Figure 30).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×31

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ vi uniqueListRaces.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ chmod 777 uniqueListRaces.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ./uniqueListRaces.sh
List of unique Races:
"Hispanic, White": 560
Asian Indian: 14
"Hispanic, Black": 24
Unknown: 23
Other: 11
Chinese: 2
"Native American, Other": 1
Hawaiian: 1
Race: 1
Black: 433
White: 4002
"Asian, Other": 18
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 30

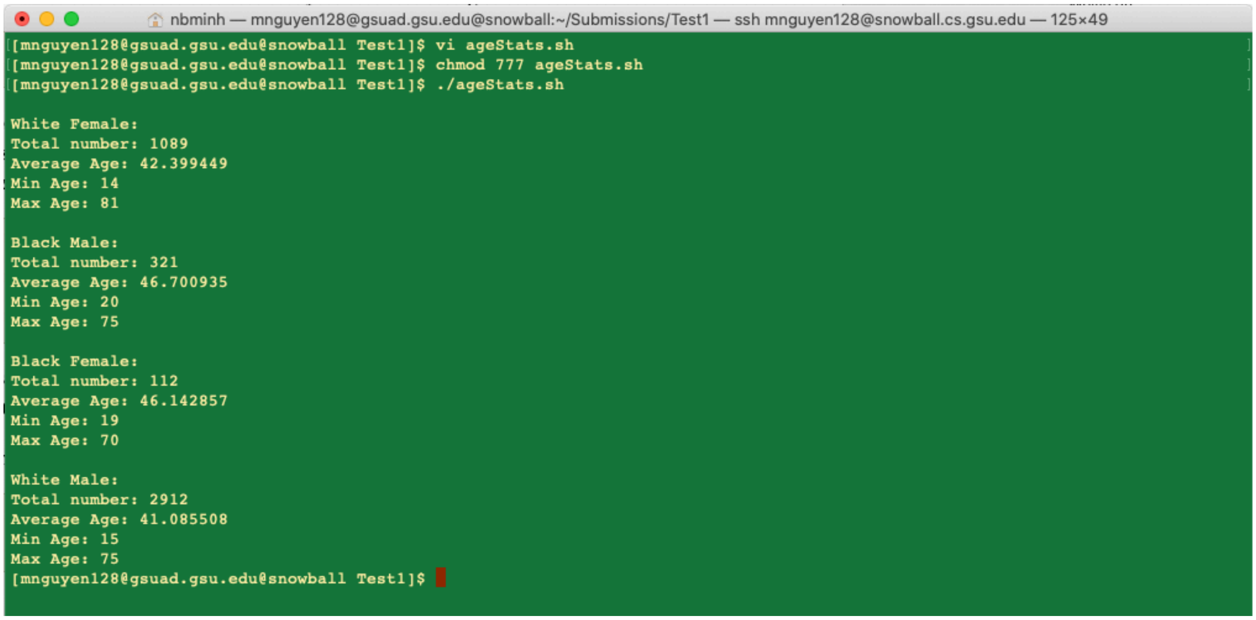
8. I wrote a script named “ageStats.sh” to find the age statistic along racial and sex (White Male, White Female, Black Male, Black Female). That includes finding the total number, average age, min age, max age. I used array (dictionary) in awk again for this question. The racial and sex become the keys for the dictionaries that are used to calculate the statistics (Figure 31).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×49

```
#!/bin/bash

awk -F '|' 'BEGIN{
count["White Male"] = count["White Female"] = count["Black Male"] = count["Black Female"] = 0;
sumAge["White Male"] = sumAge["White Female"] = sumAge["Black Male"] = sumAge["Black Female"] = 0;
minAge["White Male"] = minAge["White Female"] = minAge["Black Male"] = minAge["Black Female"] = 1000;
maxAge["White Male"] = maxAge["White Female"] = maxAge["Black Male"] = maxAge["Black Female"] = -1;
avgAge["White Male"] = avgAge["White Female"] = avgAge["Black Male"] = avgAge["Black Female"] = 0;
}
{
if (($2 == "Male" || $2 == "Female") && ($3 == "White" || $3 == "Black")) {
        str = $3 " " $2;
        count[str] += 1;
        sumAge[str] += $1;
        if (minAge[str] > $1) {
                minAge[str] = $1;
        }
        if (maxAge[str] < $1) {
                maxAge[str] = $1;
        }
}
}
END {
for (s in count) {
        printf ("\n%s:\n", s);
        printf ("Total number: %d\n", count[s]);
        if (count[s] != 0) {
                avgAge[s] = sumAge[s] / count[s];
        }
        printf ("Average Age: %f\n", avgAge[s]);
        printf ("Min Age: %d\n", minAge[s]);
        printf ("Max Age: %d\n", maxAge[s]);
}
}' 2.asr
~
```

Figure 31

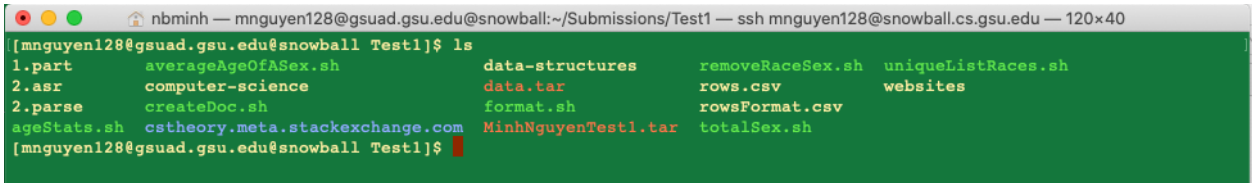
And this is the actual run of the script. Note that I changed the permissions so that I can execute the script (Figure 32).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 125×49

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ vi ageStats.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ chmod 777 ageStats.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ./ageStats.sh

White Female:
Total number: 1089
Average Age: 42.399449
Min Age: 14
Max Age: 81

Black Male:
Total number: 321
Average Age: 46.700935
Min Age: 20
Max Age: 75

Black Female:
Total number: 112
Average Age: 46.142857
Min Age: 19
Max Age: 70

White Male:
Total number: 2912
Average Age: 41.085508
Min Age: 15
Max Age: 75
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 32

9. I then saved all the files that I created and compress them all together to a tar file named “MinhNguyenTest1.tar” (Figure 33).


nbminh — mnguyen128@gsuad.gsu.edu@snowball:~/Submissions/Test1 — ssh mnguyen128@snowball.cs.gsu.edu — 120×40

```
[mnguyen128@gsuad.gsu.edu@snowball Test1]$ ls
1.part       averageAgeOfASex.sh              data-structures      removeRaceSex.sh  uniqueListRaces.sh
2.asr        computer-science                 data.tar             rows.csv          websites
2.parse      createDoc.sh                     format.sh            rowsFormat.csv
ageStats.sh  cstheory.meta.stackexchange.com  MinhNguyenTest1.tar  totalSex.sh
[mnguyen128@gsuad.gsu.edu@snowball Test1]$
```

Figure 33

The GitHub link where these files can be found is: https://github.com/mbnguyen/Test1

Part 3: Comprehension

1. What is the following bash code doing?

- curl ’http://domain.com/id/[1-151468]’ -o ’#1.html’
- grep -oh ’http://pics.domain.com/pics/original/.*jpg’ *.html >urls.txt
- sort -u urls.txt | wget -i-

Answer:

- curl ’http://domain.com/id/[1-151468]’ -o ’#1.html’

This command transfers data from a server (URL) to a file. In this case, it fetches multiple documents starting from ‘http://domain.com/id/1’ to ‘http://domain.com/id/151468’.
The outputs will be written to the files starting from ‘1.html’ to ‘151468.html’.

- grep -oh ’http://pics.domain.com/pics/original/.*jpg’ *.html > urls.txt

This command finds and matches all the contents that are links with the pattern ’http://pics.domain.com/pics/original/.*jpg’ (.jpg files). It only prints out the matched parts of a matching line. It also suppresses the prefixing of the file names on output. The input files are the .html files in the folder. Then the output lines will be written to a file named “urls.txt”.

- sort -u urls.txt | wget -i-

This command sorts the “urls.txt” file

2. Detail what the code is supposed to do in each of the 5 blocks in the vimeo_script.sh.

```
#block 1
#
#
if [ $# -ne 1 ]; then
exit 1
fi
ID=`echo $1 | awk -F / '{print $NF}'`

# Set the user agent ID to use
USER_AGENT="Mozilla/5.0"

# Check we have the tools we need
which wget
if [ $? -eq 1 ]; then
echo "ERROR: this tool requires wget on the path"
exit 1
fi
```

```
which perl
if [ $? -eq 1 ]; then
echo "ERROR: this tool requires perl on the path"
exit 1
fi
```

This block does these things:
1. It makes sure that the number arguments are parsed in is only 1 or it will exit the program.
2. Save the last part of the input (the part after the last ‘/’) to ID.
3. Set the USER_AGENT to be “Mozilla/5.0”.
4. Check if we have the tools wget and perl on the path.

```
#block 2
#
#
VIDEO_XML=`wget -U \"${USER_AGENT}\" -q -O - http://vimeo.com/${ID}`

CONFIG_URL=`echo $VIDEO_XML | grep data-config-url | perl -p -e 's/^.*? data-co$
VIDEO_CONFIG=`wget -U \"${USER_AGENT}\" -q -O - ${CONFIG_URL}`
```

This block does these things:
1. Transfer the data from the url “http://vimeo.com/${ID}” while giving the USER_AGENT to the sever.
2. Config the url to a proper format.
3. Transfer that configuration from the sever to VIDEO_CONFIG.

```
#block 3
#
#
HD_URL=`echo $VIDEO_CONFIG | perl -pe 's/^.*"hd":{(.*?)}.*$/$1/g' | perl -pe 's$
SD_URL=`echo $VIDEO_CONFIG | perl -pe 's/^.*"sd":{(.*?)}.*$/$1/g' | perl -pe 's$
CAPTION=`echo $VIDEO_XML | perl -p -e '/^.*?\<meta property="og:title" content=$
```

This block does these things:
1. Get the proper configuration url for the HD version of the video and save it to HD_URL.
2. Get the proper configuration url for the SD version of the video and save it to SD_URL.
3. Get the proper configuration url for the caption of the video and save it to CAPTION.

```
#block 4
#
#
if [ "$HD_URL" ]; then
DOWNLOAD_URL=$HD_URL
QUALITY="HD"
elif [ "$SD_URL" ]; then
DOWNLOAD_URL=$SD_URL
QUALITY="SD"
else
echo "ERROR: failed to download vimeo ID ${ID}"
echo "Please report this error at https://github.com/johnteslade/vimeo-download$
fi
```

This block does these things:
1. If we found a HD url, we put that url to the DOWNLOAD_URL with the QUALITY being “HD”.
2. If we found a SD url, we put that url to the DOWNLOAD_URL with the QUALITY being “SD”.
3. If we didn’t find any url, we show the messages.

```
#block 5
#
#
FILENAME="${CAPTION}-(${QUALITY}-${ID}).flv"
wget -U \"${USER_AGENT}\" -O ${FILENAME} ${DOWNLOAD_URL}
```

This block does these things:
1. Set a correct name of the video using CAPTION, QUALITY, ID, and the format of the video (.flv), then save that name to FILENAME.
2. Download that video from the sever using that FILENAME, the DOWNLOAD_URL while giving the sever USER_AGENT.

Part 4: Research

1. Web scrapers to pull information from website.

Web Scrapping is really interesting. I'd use the everyone's favorite tool called curl to transfer the data from the sever with a given url address. But the result of the curl command line is pretty messy and hard to read. I'd use another tool called html2text to clean the data of html codes. It then returns only the useful content without the html tags. But then, there are still the unnecessary parts of the data that needs to be removed. This time, sed command comes in handy, we can easily find the pattern of the part that we need to remove to clean up our data.

2. Email spam software that pulls emails address from websites and sends them spam emails.

The idea is pretty similar to the Web Scrapers above. I'd use the same technique to scrap the website that is given to the input. Then I'd use awk or sed to find all the emails from the data that I just collected. The patterns of a valid email could be anything followed by "@" and the known mail severs like gmail, yahoo, etc. After getting all the emails, I'd use the SendMail function to be able to send my spam mails to those emails that I got.

3. bots that grab information from live sites to make business decisions.

I'd still use the same technique as the Web Scraper to collect the data from a website. But this time, I'd use awk to find the data that I want to use. For example, if I want to find a trending product that is selling good right now, I can scrap an online product review website or a social network and find the product that a lot of people are talking about. I'd use awk to split all the words, lowercase them, count them, and put them in an array (dictionary). The words are the keys, and the numbers of times that word appears are the values of that dictionary. After that, I can easily find the most frequent words and make a business decision.

Resources:

Mester, T. (2020, February 06). Web Scraping Tutorial -- episode #1 -- Scraping a Webpage (with Bash). Retrieved July 02, 2020, from https://data36.com/web-scraping-tutorial-episode-1-scraping-a-webpage-with-bash/