```
#!/bin/bash

# In `hai.csv`, the location is split up into 3 lines,
# which is fine, but I just wanted to get rid of them for
# readability.
echo '- Getting rid of unnecessary newlines on csv (this may take a moment)...'

# Grab the header and drop it into a new csv, `temp.csv`.
head -1 downloads/hai.csv > temp.csv
# Skip the header row of `hai.csv`,
# join every 3 lines for readability (the address is split into 2 lines),
# replace 'Not Available' with 'NA' (for a smaller file size later),
# and append to `temp.csv`
tail -n +2 downloads/hai.csv | paste -d, - - - | sed -E 's/Not Available/NA/g' >>
temp.csv

# Make csv of hospitals
echo '- Extracting hospital info (this may take a moment)...'
# Extract the following columns from `temp.csv`:
#  1: Provider ID
#  2: Hospital Name
#  3: Address
#  4: City
#  5: State
#  6: ZIP Code
# 16: Location
# Then sort the csv by Provider ID,
# filter for uniques,
# and create a new csv, `hospitals_temp.csv`.
csvcut -c 1,2,3,4,5,6,16 temp.csv | csvsort -c 1 | uniq > hospitals_temp.csv

# Make csv of footnotes
echo '- Filtering rows with footnotes...'
# Grep for the central line data that we care about (skipping confidence intervals),
# cut out columns 1 (Provider ID) and 13 (Footnotes),
# grep Footnotes column (which is now column 2) for cells with numbers (footnotes),
# sort Footnotes column,
# filter for uniques,
# reverse grep for footnote about confidence intervals because we don't need that,
# and create a new csv, `footnotes_temp.csv`.
csvgrep -c 10 -r "HAI_1_DOPC_DAYS|HAI_1_NUMERATOR|HAI_1_SIR" temp.csv | csvcut -c 1,13 |
csvgrep -c 2 -r '\d+' | csvsort -c 2 | uniq | csvgrep -c 2 -i -m "8 - The lower limit of
the confidence interval cannot be calculated if the number of observed infections equals
zero." > footnotes_temp.csv
# (Note to self: Once you filter out the measures, there's no `3, 8` combo footnote.)

# Return a list of footnotes
echo '- Creating a list of footnotes (this may take a moment)...'
# Extract the Footnotes column in `footnotes_temp.csv`
# but not the header row,
# sort for uniques,
# and create `footnotes_list.txt`
csvcut -c 2 footnotes_temp.csv | tail -n +2 | sort -u > footnotes_list.txt

# Make the data a bit more readable by cutting out the footnotes' long explanations.
# We'll place the results in `footnotes.csv`.
```

```
53 echo '- Shortening footnotes...'
54 sed -E 's/12 - This measure does not apply to this hospital for this reporting
   period./12/g' footnotes_temp.csv | sed -E 's/13 - Results cannot be calculated for this
   reporting period./13/g' | sed -E 's/3 - Results are based on a shorter time period than
   required./3/g' | sed -E 's/5 - Results are not available for this reporting period./5/g'
   | sed -E 's/8 - The lower limit of the confidence interval cannot be calculated if the
   number of observed infections equals zero./8/g' | sed -E 's/, /,/g' > footnotes.csv
55
56
57 echo '- Formatting latitude and longitude...'
58 # The hospital data comes with geographic coordinates for each hospital,
59 # formatted like `(31.21537937900007, -85.36146587999997)`.
60 # I very likely did not need a separate Python script for this,
61 # and may not have needed to split the coordinates at all.
62 python location.py
63 # `location.py` outputs `hospitals_info.csv`.
64
65 echo '- Filtering hospital type info...'
66 # `hospital_general.csv` has the same issue as `hai.csv` in that
67 # the location has a couple newlines in it. Not going to join lines
68 # for readability here since the following will extract columns
69 # without newlines.
70 # Extract the following columns out of `downloads/hospital_general.csv`:
71 #  1: Provider ID
72 #  9: Hospital Type
73 # 10: Hospital Ownership
74 # 11: Emergency Services
75 # and put them in `hospitals_type.csv`.
76 csvcut -c 1,9,10,11 downloads/hospital_general.csv > hospitals_type.csv
77
78 echo '- Joining hospital general information data...'
79 # Left join `hospitals_info.csv` and `hospitals_type.csv` on their first columns
   (Provider ID),
80 # remove the 9th column because that's a dupe of Provider ID,
81 # and put it all in `hospitals.csv`.
82 csvjoin -c 1,1 hospitals_info.csv hospitals_type.csv | csvcut -C 9 > hospitals.csv
83 # Note: There are more rows in `hospitals_type.csv` than `hospitals_info.csv`
84 # because not all hospitals report central line measures to this particular agency.
85
86 echo '- Filtering out CLABSI measures (this may take a moment)...'
87 # Grep for rows with one of the central line measures in the 10th column,
88 # then extract the following columns:
89 #  1: Provider ID
90 # 10: Measure ID
91 # 12: Score
92 # and dump that all in `clabsi_temp.csv`
93 csvgrep -c 10 -r "HAI_1_DOPC_DAYS|HAI_1_NUMERATOR|HAI_1_SIR" temp.csv | csvcut -c 1,10,12
   > clabsi_temp.csv
94
95 echo '- Filtering CLABSI SIR...'
96 # Grep for the SIR rows,
97 # remove the label column,
98 # and dump to `clabsi_sir.csv`.
99 csvgrep -c 2 -m "HAI_1_SIR" clabsi_temp.csv | csvcut -C 2 > clabsi_sir.csv
100
101 echo '- Filtering CLABSI days...'
102 # Grep for the days rows,
```

```
# remove the label column,
# and dump to `clabsi_days.csv`.
csvgrep -c 2 -m "HAI_1_DOPC_DAYS" clabsi_temp.csv | csvcut -C 2 > clabsi_days.csv

echo '- Filtering CLABSI observed cases...'
# Grep for the observed rows,
# remove the label column,
# and dump to `clabsi_observed.csv`.
csvgrep -c 2 -m "HAI_1_NUMERATOR" clabsi_temp.csv | csvcut -C 2 > clabsi_observed.csv

# Join the data we need into one big table.
echo '- Joining tables...'

# Write a header row to `hospitals_clabsi.csv`.
echo
'provider_id,hospital_name,street,city,state,zip_code,lat,lng,type,ownership,emergency_se
rvices,observed,days,sir,footnotes' > hospitals_clabsi.csv

# Left join on the "Provider ID" column: `hospitals.csv`, the csvs with all the CLABSI
scores, and the footnotes,
# remove the columns that are dupes of "Provider ID" after the join,
# and stream everything (but the header row) into `hospitals_clabsi.csv`.
csvjoin -c "Provider ID" --left hospitals.csv clabsi_observed.csv clabsi_days.csv
clabsi_sir.csv footnotes.csv | csvcut -C 12,14,16,18 | tail -n +2 >> hospitals_clabsi.csv

echo '- Cleaning up data...'

# Does everything look good?
csvclean hospitals_clabsi.csv

# It does.
mv hospitals_clabsi_out.csv hospitals_clabsi.csv

echo '- Removing unnecessary files...'
rm clabsi_days.csv clabsi_observed.csv clabsi_sir.csv clabsi_temp.csv hospitals.csv
hospitals_info.csv hospitals_temp.csv hospitals_type.csv footnotes_temp.csv footnotes.csv
temp.csv

echo 'Data processing complete. Check `hospitals_clabsi.csv` for complete table, and
`foonotes_list.txt` for a list of what the footnotes are.'

```