Jared Yu (914640019)

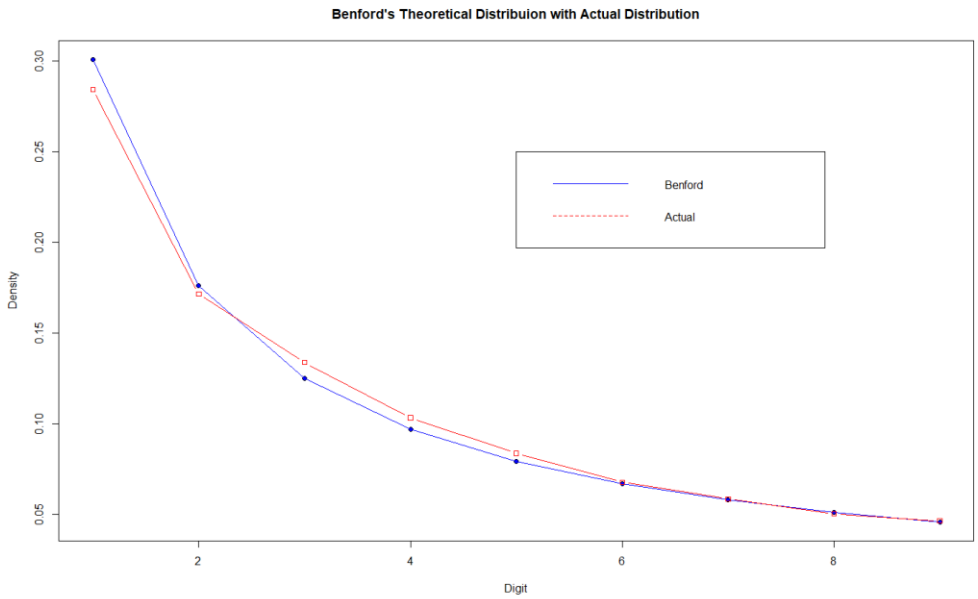
Questions

1. The following is a list of steps that was used to find the solutions:
    1. Subset the data using the cluster.
        a. The entire code can be found in the appendix (hw5.sh), but it involved the commands `cut`, `sort`, and `uniq`. The code to run the shell script can be found in `submit.sh`.
        b. The data was written into a file called `unique_cols.csv`, and was retrieved from the cluster by using `sftp` to login to the cluster and downloading it with the command `get`.
    2. Load the data into RStudio for further filtering.
        a. The data was then read into RStudio, into an R script called `bootstrap.R` and the following filtering was done:
            i. Keep only columns `total_obligation` and `parent_recpient_unique_id` ignoring `action_date`.
            ii. Remove NA's from `total_obligation`.
            iii. Remove values smaller than 1.
            iv. Remove `parent_recpient_unique_id` that have frequencies less than 100.
            v. Subset on the first digit of the values in `total_obligation`.
        b. Write the filtered data to a file called `bootstrap.csv`.
    3. The filtered data was then copied onto the cluster using the command `scp`. Additionally, an R script called `bootstrap_code.R` was used both on the local machine and on the cluster. It was run on cluster using a shell script called `submit_R.sh`. The file does the following:
        a. Create a list of observations based on the `parent_recpient_unique_id`.
        b. Create a table that finds the counts of digits 1-9 for each recipient.
        c. Find the overall distribution of digits for the entire dataset, this is known as $Q(x)$ in Kullback-Leibler Divergence (KLD). Also, find the distribution of digits for each recipient, this is known as $P(x)$ in KLD.
        d. Find the KLD per recipient.
        e. Bootstrap 1,000 times per recipient and compute the quantiles.
        f. Save the quantiles along with the actual KLD for all the recipients, written to a file called `KLD_ci.csv`.

2. The following is a table of the digits 1-9 and their distribution for the entire set.

1	2	3	4	5	6	7	8	9
0.284	0.172	0.134	0.103	0.084	0.068	0.059	0.050	0.046

3. Below is a plot of both the distribution of digits 1-9 for Benford's theoretical distribution and the actual distribution of the first digits from the dataset:



The two distributions share a highly similar shape, where the smaller digits have a higher frequency in comparison to the larger digits. In the Bedford's theoretical distribution, it is apparent that there is more of a skew, as the 1's have a higher frequency in comparison to the dataset. This result is not surprising, since this is a real-life set of naturally occurring numerical data. Benford's law says that it is likely that the leading digit is small. In this case, the 1's are quite close to 30% and 9's are quite close to 5%, just as stated in Benford's law.

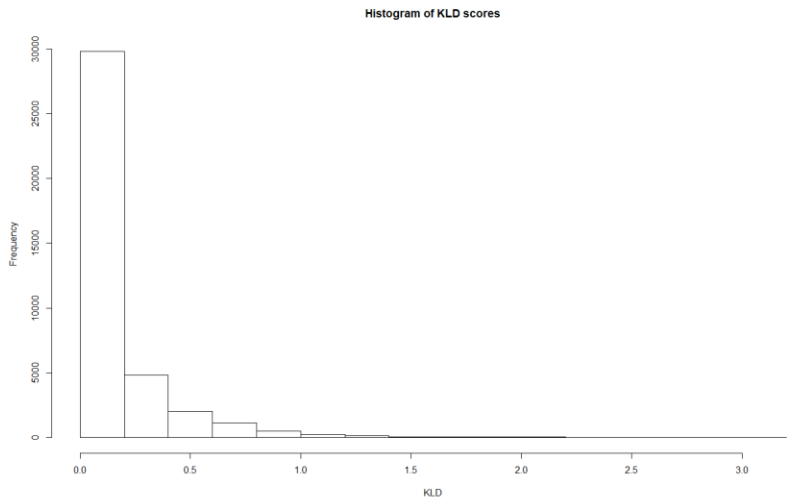
When using the actual distribution as P, and the Benford formula for Q, the $D_{KL}(P,Q) = 0.001189176$. When using the uniform distribution for P, and the same Q as before, the $D_{KL}(P,Q) = 0.1912054$. It makes sense that the former value is much smaller, as it is apparent from the two plots that the two values follow each other highly close, and look almost identical in terms of their shape. The latter is a much larger number, as it is apparent that the appearance of a horizontal uniform distribution and that of Benford's theoretical distribution are quite different. One has a strong skew, while the other is a flat line.

4. Looking at the length different recipients, there are 38,818 different recipients. One of these are blank, representing observations where the funding recipient was NA. Another recipient included the ID `000000000`. Additionally, there was a non-numeric ID called `INMARSAT`. So not including the blank recipient category, there were 38,817 different recipients.

   The following are some summary statistics for the KLD scores:

Mean	*Median*	*Mode*	*Max*	*Min*	2.5%	97.5%
0.162488	0.077401	1.257215	3.069721	0.000147	0.005489019	0.822134830

   The mean and the median are not too close, with the mean being more than twice the size of the median. It is logical that they are not close, because the distribution of the KLD scores are quite skewed. This is evident in the following histogram:



The maximum value is a significant outlier, extending well beyond the 97.5th percentile. The minimum value is small, but there are many numbers which are close to 0. The quantiles are also given, showing the range where 95% of the data lie within.

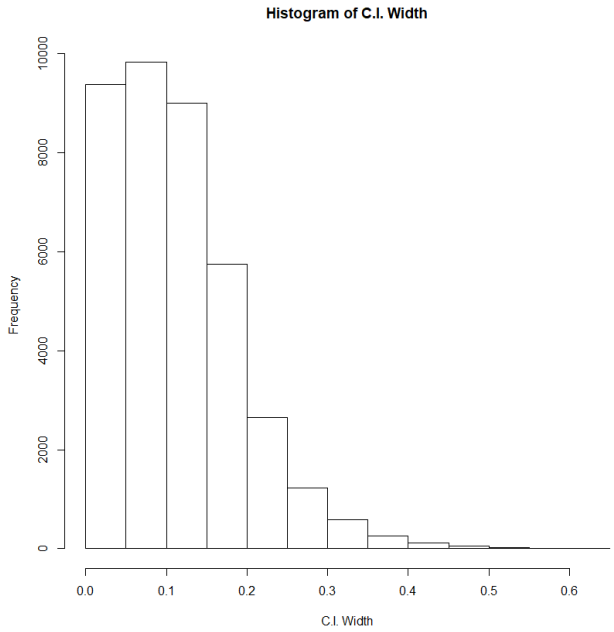
5. Using the `which.max()` function, it is possible to find the largest KLD value which is 3.069721, along with the recipient. There are three with this value, and a total of 20 recipients with a KLD larger than 2.5. One of the three were arbitrarily chosen, and the recipient's ID is 007836612. To find out information on the transactions, `hw5b.sh` was run in the cluster. The code can be found in the appendix. The way to find transactions for this recipient was done primarily with the bash command `grep`. The data was written to a file called `kld_large.csv`. The transactions seem to all be for a modernization project for the Internal Revenue Service in Andover, Massachusetts. Looking at the location on Google Maps, it seems that this is a rather large building which has been heavily invested in. Apparently, the reason is because of the American Recovery and Reinvestment Act (ARRA) of 2009 that all the money has been granted for construction of this IRS campus.

Bootstrap

1. Below are summary statistics for the confidence intervals widths of the KLD from bootstrapping.

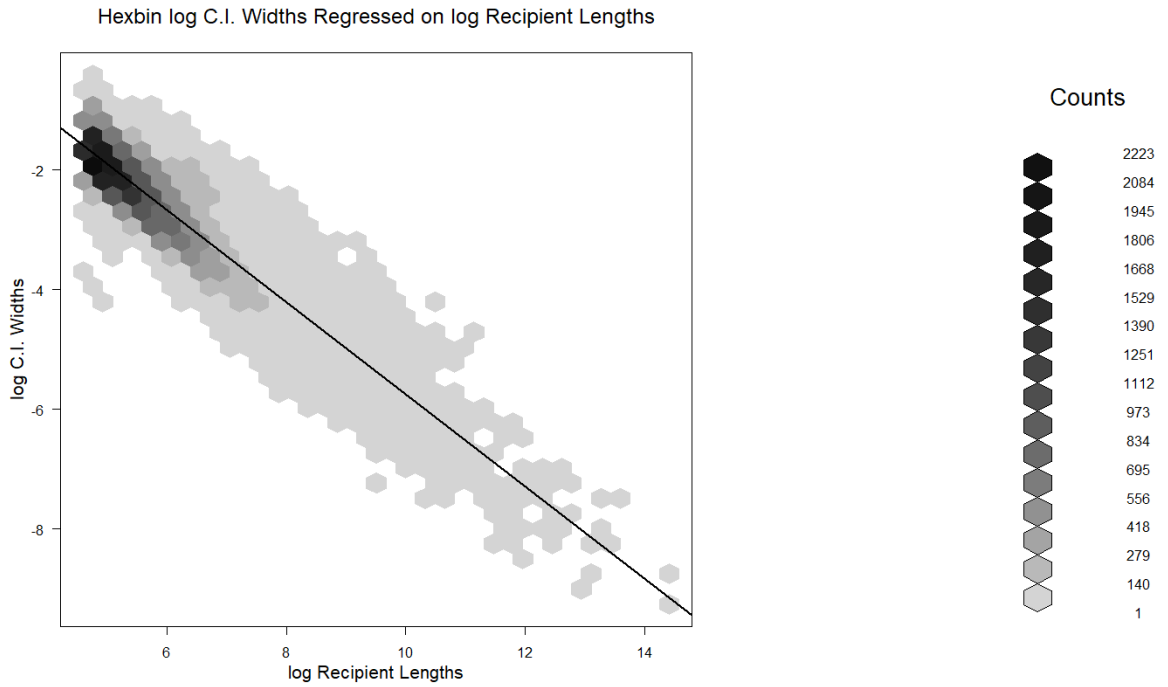
$Mean$	$Median$	$Mode$	Max	Min	2.5%	97.5%
0.11244	0.10096	0	0.60227	0	0.006423902	0.302323703

An interesting note is that the mode and min show that there are several widths of size 0. After some investigation, it is apparent that this comes from recipients where all the digits are the same, for example all 1's or all 9's. When this is the case, the quantiles that are calculated are the same number, and so the widths must also be 0. Below is a histogram of the widths of the confidence intervals:



The histogram is quite skewed, with many of the widths being close in size to 0, indicating that the widths in general are quite narrow. However, this is not surprising, considering that the actual KLD values themselves are also rather small and close to 0. The narrow widths is a good thing, indicating that the bootstrap confidence intervals are good at predicting their respective KLD values. The tighter the widths, the more robust the bootstrap process. It is interesting to note that despite the confidence level being set to $\alpha = 0.05$, there are still 104 recipients where the actual KLD value is below the lower bound of the confidence interval. This number is less than 1% of the total.

2. The goal is to understand the relationship between the widths of the confidence intervals and the size of each recipient. The idea is to treat the width as a function of the size of the recipient, so the goal is to use linear regression for this task. To get a better understanding, the log of the widths was regressed on the log of the recipient sizes. Below is an image of the plot and the regression:



To be able to find the fit, widths which were 0 have been removed. Additionally, the NA recipient was not included as well, it is a considerable large outlier. The Hexbin is a plot which helps to better understand the density of the points, since there are almost 40,000 recipients, it is difficult to get a solid understanding with just a simple scatter plot. The plot shows that as the length of the log recipients increases, the log of the confidence interval widths will decrease. The interpretation then is that a percentage increase in length of recipient leads to a -0.3329-percentage decrease in the width of the confidence interval.

Code Appendix

hw5.sh

```
COLUMNS_TO_SUBSET=3,8,52
unzip -p ${DATAFILE} |
        cut --delimiter=, --fields=${COLUMNS_TO_SUBSET} |
        sort --reverse |
        uniq |
        cat > unique_cols.csv
```

submit.sh

```
#!/bin/bash -l
# Use the staclass partition. Only applies if you are in STA141C
#SBATCH --partition=staclass
# Use two cores to get some pipeline parallelism
#SBATCH --ntasks=2
# Give the job a name
#SBATCH --job-name=hw5
#SBATCH --mail-type=ALL
#SBATCH --mail-user=qzyu@ucdavis.edu
export DATAFILE="/scratch/transaction.zip"
bash hw5.sh
```

submit_R.sh

```
#!/bin/bash -l
# Use the staclass partition. Only applies if you are in STA141C
#SBATCH --partition=staclass
# Use two cores to get some pipeline parallelism
#SBATCH --ntasks=1
# Give the job a name
#SBATCH --job-name=boot

#SBATCH --mail-type=ALL
#SBATCH --mail-user=qzyu@ucdavis.edu
module load R
Rscript bootstrap_code.R
```

hw5b.sh

```
COLUMNS_TO_SUBSET=25,52
KEY_WORD="007836612"
unzip -p ${DATAFILE} |
```

```
cut --delimiter=, --fields=${COLUMNS_TO_SUBSET} |
grep --ignore-case ${KEY_WORD} |
cat > kld_large.csv
```

Reference

Haoran Zhang

https://www.computerhope.com/unix/uuniq.htm

https://www.rdocumentation.org/packages/base/versions/3.5.2/topics/table

https://www.statmethods.net/management/sorting.html

https://stackoverflow.com/questions/11254524/omit-rows-containing-specific-column-of-na

https://askubuntu.com/questions/101587/how-do-i-enter-a-file-or-directory-with-special-characters-in-its-name

https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/

https://github.com/clarkfitzg/slurm-example

https://stat.ethz.ch/R-manual/R-devel/library/base/html/substr.html

https://stackoverflow.com/questions/21675379/r-only-keep-the-3-x-first-characters-in-a-all-rows-in-a-column

https://stat.ethz.ch/R-manual/R-devel/library/base/html/colSums.html

https://stackoverflow.com/questions/6432067/sequence-of-repeated-values-in-r

https://www.dummies.com/programming/r/how-to-repeat-vectors-in-r/

https://stackoverflow.com/questions/27962282/how-to-reset-row-names

https://stackoverflow.com/questions/24831580/return-row-of-data-frame-based-on-value-in-a-column-r

https://www.statmethods.net/graphs/line.html

https://en.wikipedia.org/wiki/Benford%27s_law

https://www.tutorialspoint.com/r/r_mean_median_mode.htm

https://www.google.com/maps/@42.6484698,-71.1832498,3a,60y,112.12h,86.21t/data=!3m6!1e1!3m4!1skQ2fK7dsME2kLt5ryitXAw!2e0!7i13312!8i6656

https://govtribe.com/opportunity/federal-contract-opportunity/recovery-dot-modernization-of-the-irs-service-center-in-andover-ma-gs01p09bzc0014

https://www.statmethods.net/management/sorting.html

https://stackoverflow.com/questions/8519998/find-rows-in-a-data-frame-where-two-columns-are-equal

https://stat.ethz.ch/pipermail/r-help/2003-June/035433.html

http://home.wlu.edu/~gusej/econ398/notes/logRegressions.pdf

https://stackoverflow.com/questions/4881930/remove-the-last-line-from-a-file-in-bash

https://rdrr.io/cran/hexbin/man/hexVP.abline.html