HW02 — STAT/CS 287  
NAME: Michael Arnold  
DATE: 2018-09-27

## P1.1

In the HW02 directory there exists a python script, this report document, and a directory called reports, which contains the data files. The Python script only uses the working directory by taking files from /reports/\* rather than the full path.

## P1.2

The function of load\_reports works by opening a file and for each line in the file splits on a colon. The colon was chosen since there is an arbitrary number of white space between the error mode and the counts. This split line is then sent to a try except block, which attempts to cast the last element of the list to an int. If this throws no errors, we set the dictionary key to the error type, and the value to the integer count of the errors. After looping over the file, the dictionary is returned. If a line fails to cast the last split element to an integer, the ValueError is excepted, since there is commonly a colon separating the date of the report, and a descriptor, which is not an error count.

## P1.3

The loop calling the function simply uses glob.glob, to expand the reports directory, and generates a list of all files inside. This list of file names is given to the load report function. This function returns a dictionary, which is placed in a master dictionary global\_errors, as a value. The key associated with this value is simply the string id number pulled from the filename. I know that the load report function works correctly, because all 1431 files are assigned a key in the dictionary, which has length 1431.

## P1.4

We know the ID’s are unique since they are the keys in a dictionary. I wrote a loop to check if there were any non-sequential data centers, and there were none missing. I also checked to see that the filename was the same as the DATACENTER ID on the first line of every file. The function check\_id will print “error” if they don’t match, but when executed on all files, it doesn’t print anything. Thus I concluded the ID’s are consistent.

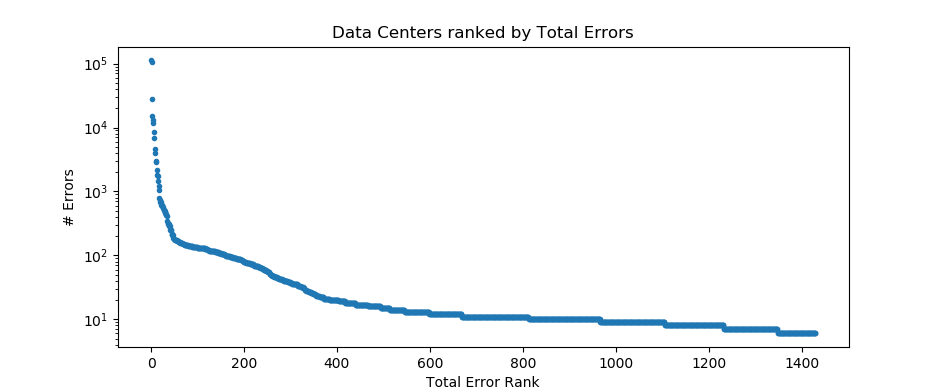
## P1.5

I made the error modes consistent by mapping the “A/C” and “Air Cond.” error modes to the “HVAC” error mode, since these were just different names for the same thing, and each entry had only one of these three choices. I made a function to return the set of all unique error modes, which showed these repeated categories, and then used it to confirm they were all merged after merging.

## P1.6

For this question I just made sure that the values make sense. We already made sure the data could be cast to an int type. I additionally made sure that all the values were positive, and looked at the max and min. Strangely, there were no employee errors, and power losses.

## P2.1

Yes, some data centers experience drastically more errors than others. 000223 had the most at 111967 errors, and probably needs a shake up. Hopefully these errors aren’t costly. There were a bunch with only 6 errors at the other end.

## P2.2

The average number of errors a data center should expect is about 262 during this period. However, this average is heavily skewed by the high error Data Centers. The median number of errors is only 11. The mode was 10.

## P2.3

The most common type of error was the Mics. Elec. Which makes sense, since there are presumably many electrical things which could go wrong at a data center.

## P2.4

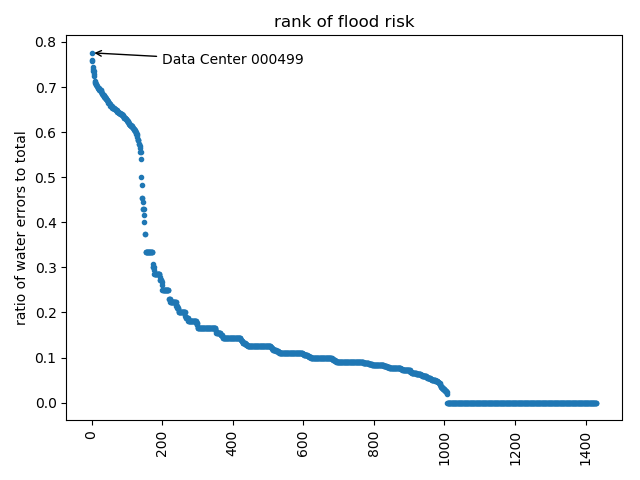
The riskiest error modes are the ones which contain the most risk of shutting down the data center, and thus must be ones which are the least likely to occur, since the company doesn’t make money when the data center is shut down. Power/generator loss will shutdown the data center, so if that event occurs there is a 100% chance of the center shutting down.

However, if we want to define risk as the probability of the event shutting down the center times the probability of the event occurring, then we might say that Flooding or HVAC is the most risky, where there is a high frequency of occurrence, and also a high probability of shutdown.

In making both judgements I’ve assumed that the probability of Misc. electrical errors causing a shutdown is small, since they happen all the time, and this company wouldn’t be around for 2 whole years if they spent most of the time shutdown.

To make an accurate ranking of risk, we’d need a history of shutdowns by error cause, in addition to the frequency of events occurring.

## P3.1

I made a ranking, and plot of that ranking reflected below.

## P3.2

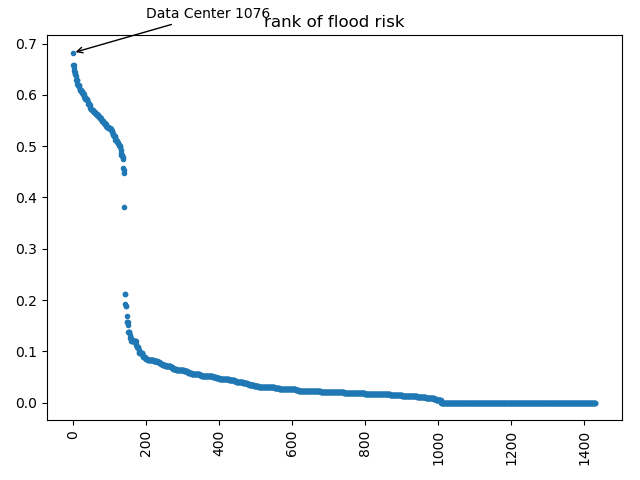
## :qP3.2 Bonus

The best way to calculate the uncertainty in this case would be to find the 95% confidence interval around our ratio, given that the number of floods and the total number of errors. We can calculate this with the Wilson confidence interval. Using the lower bound of the confidence interval we can construct a new ranking, that will rank data centers which have a higher total number of errors as more risky than a data center with the same flood ratio, but a very small number of errors overall.

In this way we rank a data center which had 50 floods errors out of 100 total errors as a higher flood risk, and a center with one flooding event, but only 2 total errors.

Actually doing the ranking without uncertainty we see the top ranked data center 000499 has only 58 total errors, while with uncertainty, the riskiest data center is 001076, with 141 total errors. The distribution also separates visually after including uncertainty.

## P3.3 Bonus

  
Illustration 1: Including uncertainty

If we wanted to have a statistic that showed how the distribution as a whole changed after we incorporated uncertainty into our ranking, we could use Jensen–Shannon divergence. This measure allows to quantify similarity of the ranking as a whole, but doesn’t give us a lot of information about how individual data centers fell or rose in relation to each other.