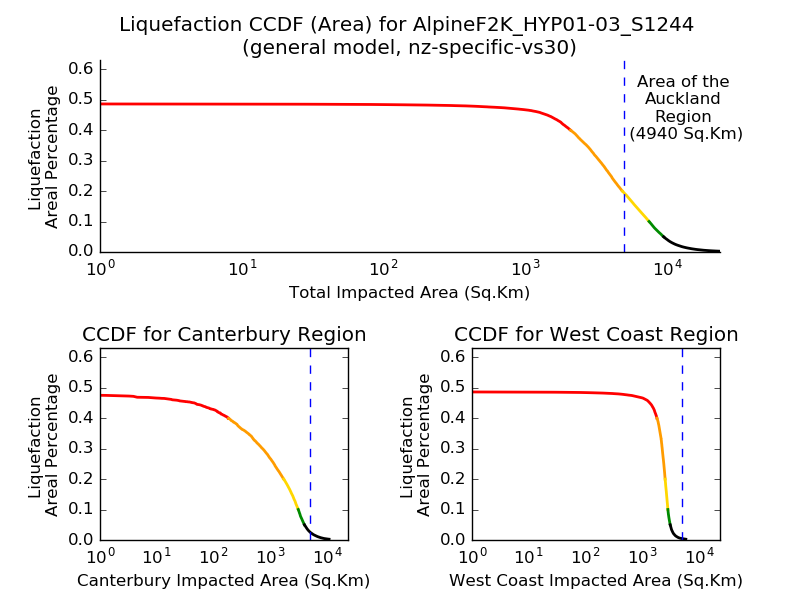
**Documentation**

**CCDF\_regional\_popn.py**

This script takes any xyz file where each line holds longitude, latitude, probability data in that order, and produces a CCDF of areal probability against cumulative area/population/dwellings. As the script works it prints the directory where the figure will be saved, the name of the file, the total impacted area/population/dwellings, and a “Done” message. If the script encounters an error, then it will display a message with a recommended fix before ending the script.

**Example output:**



**Usage:** python CCDF\_regional\_popn.py /path/to/xyz --title --xAxis --datatype --regions --save  
  
**Example:** python CCDF\_regional\_popn.py /home/nesi00213/RunFolder/Cybershake/v17p8/Runs/AlpineF2K/Impact/Liquefaction/AlpineF2K\_HYP01-03\_S1244/AlpineF2K\_zhu\_2016\_general\_probability\_nz-specific-vs30.xyz --title This is a CCDF --xAxis pop --datatype liq --regions off --save here  
  
**Optional arguments:**

* **-t, --title:** Add this argument followed by a title for the figure. If no title is given, then the script will try to auto-generate one from the path to the xyz file. If the title cannot be generated, then an error message will appear, and the script will end.
* **-x, --xAxis:** Use this to change what is plotted on the x-axis. “--xAxis pop” will plot population on the x-axis, “--xAxis dwell” will plot dwellings on the x-axis, and leaving it out will plot area on the x-axis.   
  **Note:** population and dwelling data is currently unreliable (see popn\_db.txt and dwelling\_db.txt documentation)
* **-d, --datatype:** Use this to specify whether the areal probability data is liquefaction or landslide. If the argument isn’t given, then the script will try to find the datatype from the path to the xyz file. Recognised arguments: “--datatype liq”, “--datatype liquefaction”, “--datatype ls”, and “--datatype landslide”
* **-r, --regions:** Use to turn off region subplots. By default, the final figure will show a total CCDF at the top and two CCDFs for the two most impacted regions. “--regions off” will remove the subplots for the regions and produce a single total CCDF.
* **-s, --save:** Use “--save here” to save the figure to the current working directory. By default, the figure will save to a CCDF directory within the directory that the xyz file is saved to. If this CCDF directory does not exist, then an error message will appear and end the script.

**Important:**

* The landslide bins are currently not functioning properly. The very high bin should be 100% to 9%, high bin from 9% to 4%, moderate bin from 4% to 1%, low bin from 1% to 0.25%, and very low from 0.25% to 0%. Since the landslide zero cut off is 0.5%, the very low bin would be totally excluded. To work around this, I’ve temporarily set up the bins to be: very high from 100% to 9%, high from 9% to 4%, moderate from 4% to 2.5%, no low bin, and very low from 2.5% to 0.5%.

**Notes:**

* There are three text files required for full functionality of the script:   
   Region\_database.txt  
   popn\_db.txt  
   dwelling\_db.txt  
  Please see the documentation on these files for further information
* The filename of the final figure is based on the name of the xyz file, and which optional arguments are provided
* Takes between 5 and 45 seconds to run (Depends on which optional arguments are given, and the size of the xyz file)
* The script assumes that if there is any data in the very high bin then there must be data in the high bin, if there is data in the high bin then there must be data in the moderate bin, and so on
* The script is currently saved in /home/fordw/GroundFailure/scripts/

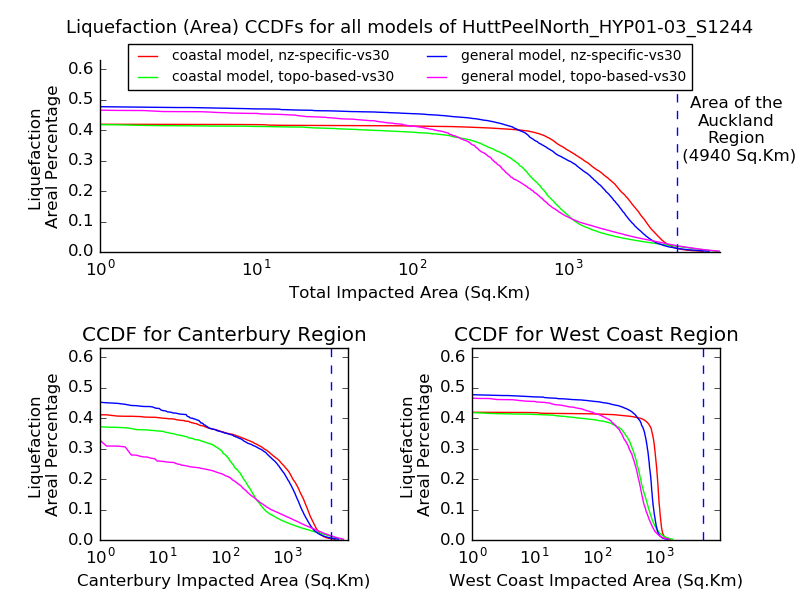
**Next steps:**

* The population and dwelling data needs to be validated (currently I do not believe it is reliable)
* In future, reference lines for population and dwellings could be added

**All\_models\_popn.py**

This script takes a path to a directory which holds multiple xyz files of longitude, latitude, liquefaction probability data, and produces multiple CCDFs of liquefaction areal probability against cumulative area/population/dwellings. It is designed to plot 4 different liquefaction probability models at once. The script checks the given directory for certain xyz files which correspond to the different liquefaction models (find in the MODELS constant). As it works, the script prints the directory that the figure will save to, the file name, the maximum impacted area/population/dwellings, and a “Done” message. If the script encounters an error, then it will display a message with a recommended fix before ending the script.

**Example output:**



**Usage:** python All\_models\_popn.py /path/to/directory/of/xyz/files --title --xAxis --save --regions

**Example:** python All\_models\_popn.py /home/nesi00213/RunFolder/Cybershake/v17p9/Runs/HuttPeelNorth/Impact/Liquefaction/HuttPeelNorth\_HYP01-03\_S1244 -t Title test -x pop -s here -r off

**Optional Arguments:**

* **-t, --title:** Add this argument followed by a title for the figure. If no title is given, then the script will try to auto-generate one from the path to the xyz file. If the title cannot be generated, then an error message will appear, and the script will end.
* **-x, --xAxis:** Use this to change what is plotted on the x-axis. “--xAxis pop” will plot population on the x-axis, “--xAxis dwell” will plot dwellings on the x-axis, and leaving it out will plot area on the x-axis.   
  **Note:** population and dwelling data is currently unreliable (see popn\_db.txt and dwelling\_db.txt documentation)
* **-r, --regions:** Use to turn off region subplots. By default, the final figure will show a total CCDF at the top and two CCDFs for the two most impacted regions. “--regions off” will remove the subplots for the regions and produce a single total CCDF.
* **-s, --save:** Use “--save here” to save the figure to the current working directory. By default, the figure will save to a CCDF directory within the directory containing the xyz files. If this CCDF directory does not exist, then an error message will appear and end the script.

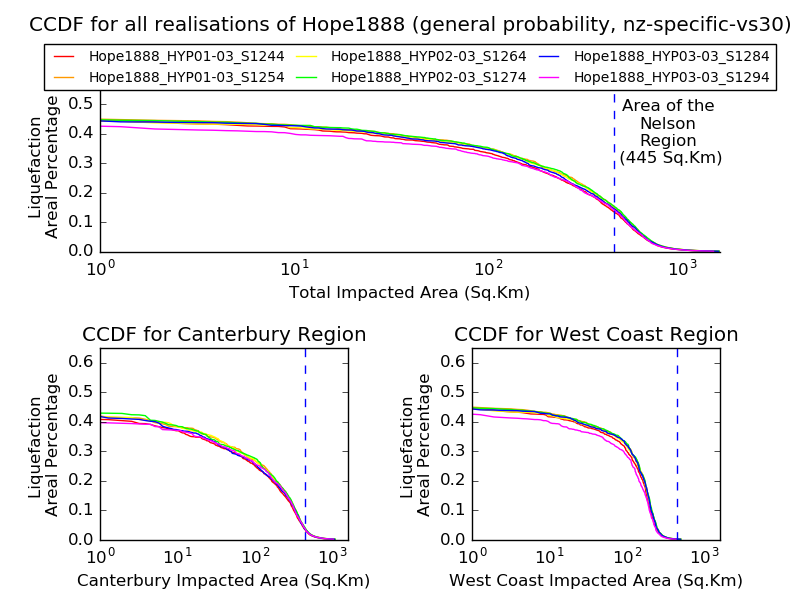
**Notes:**

* There are three text files required for full functionality of the script:   
   Region\_database.txt  
   popn\_db.txt  
   dwelling\_db.txt  
  Please see the documentation on these files for further information
* This script only reads “liquefaction areal percentage” on the y-axis because (at the time this was made) there was only one landslide probability model and so this script was not designed to work for landslide data
* The script takes between 5 and 45 seconds to run, depending on data given and the arguments (in general using population or dwelling data adds 30 seconds to the runtime)
* Assumes that all models have the same cell area and so the cell area is only calculated once. If using models with different areas then update the cell area for each model
* The script is currently saved /home/fordw/GroundFailure/scripts/
* **Next steps:**
* When new liquefaction models are introduced, simply change the MODELS constant and, if necessary, add extra plotting colours to the COLOURS constant.
* The population and dwelling data needs to be validated (currently I do not believe it is reliable)
* In future, reference lines for population and dwellings could be added

**All\_realisations\_popn.py**

This script takes the path to a run on Hypocentre, finds an xyz file for each realisation, and plots the areal probability against area, population, or dwellings for each realisation. The output is a CCDF comparing all the realisations for a run. The script outputs the directory that the figure will be saved to, the name the figure is saved as, and the maximum impacted area/pop/dwellings impacted from one realisation.

**Example output:**



**Usage:** python All\_realisations\_popn.py /path/to/run --datatype --save --xAxis --mean --regions

**Example:** python All\_realisations\_popn.py /home/nesi00213/RunFolder/Cybershake/v17p8/Runs/Hope1888 --datatype liq --save here --xAxis pop --mean on --regions off

**Optional arguments:**

* **-d, --datatype:** Use this to specify whether the areal probability data to plot is liquefaction or landslide. If the argument isn’t given, then the script will default to use liquefaction data. Recognised arguments: “--datatype liq”, “--datatype liquefaction”, “--datatype ls”, and “--datatype landslide”
* **-s, --save:** Use “--save here” to save the figure to the current working directory. By default, the figure will save to a CCDF directory within the /path/to/run/Impact/Liquefaction/ or /path/to/run/Impact/Landslide/ directory. If this CCDF directory does not exist, then an error message will appear and end the script.
* **-x, --xAxis:** Use this to change what is plotted on the x-axis. “--xAxis pop” will plot population on the x-axis, “--xAxis dwell” will plot dwellings on the x-axis, and leaving it out will plot area on the x-axis.   
  **Note:** population and dwelling data is currently unreliable (see popn\_db.txt and dwelling\_db.txt documentation)
* **-m, --mean:** Use “--mean on” to add a line for the mean of all the realisations to the figure. The mean line will plot in black and dashed standard deviation lines will also be plotted in black. All other lines will plot in grey and the legend will be removed. Works for region subplots as well.  
  **Note:** Because the x-axis data isn’t the same for each realisation, it is difficult to find the mean line for population/dwelling data. All lines plot in grey with no mean line when “--mean on” and “--xAxis pop/dwell”
* **-r, --regions:** Use to turn off region subplots. By default, the final figure will show a total CCDF at the top and two CCDFs for the two most impacted regions. “--regions off” will remove the subplots for the regions and produce a single total CCDF.

**Notes:**

* The legend becomes too large when trying to plot more than 6 realisations. To work around this, the script will plot all realisations in grey (with no legend) if more than 6 realisations need plotting.
* There are three text files required for full functionality of the script:   
   Region\_database.txt  
   popn\_db.txt  
   dwelling\_db.txt  
  Please see the documentation on these files for further information
* This script only reads “liquefaction areal percentage” on the y-axis because (at the time this was made) there was only one landslide probability model and so this script was not designed to work for landslide data
* The script can take several minutes to process landslide data
* Assumes that all realisations have the same cell area and so the cell area is only calculated once. If the realisations have different cell areas, then update the cell area for each realisation.
* The script is currently saved to /home/fordw/GroundFailure/scripts/

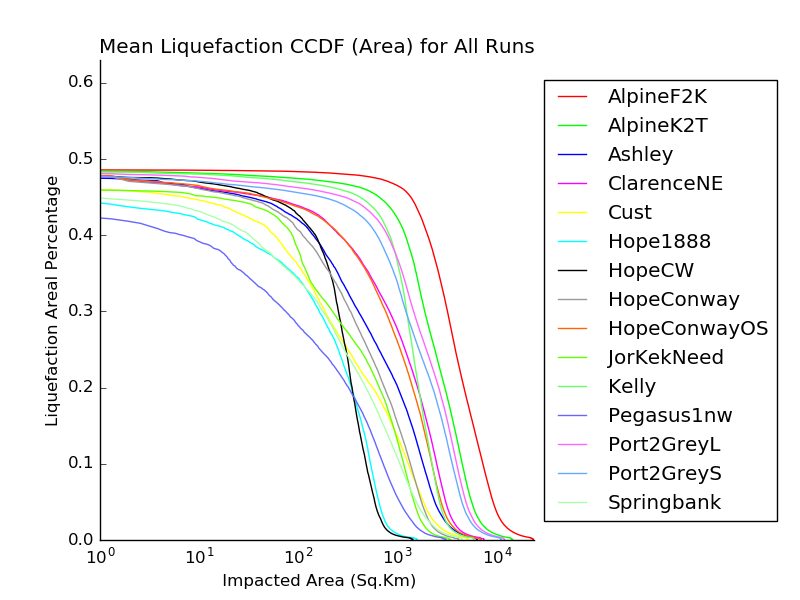
**Next steps:**

* When new liquefaction and landslide models are introduced, simply change the LIQ\_MODEL and/or LS\_MODEL constants
* The population and dwelling data needs to be validated (currently I do not believe it is reliable)
* In future, reference lines for population and dwellings could be added
* Could investigate a way to plot a mean line for the population/dwelling data
* It could be useful to plot specific realisations in colour, with other realisations greyed out (like the All\_runs\_popn.py script)

**All\_runs\_popn.py**

This script plots a CCDF of areal probability against area, population, or dwellings for multiple runs. The script plots the mean of all realisations for a run. The real power of the script is the range of optional arguments available. As the script runs it will print the directory that the figure will save to, each run that has been completed (to track progress), the final path to the figure, and the maximum area/pop/dwellings impacted by a run.

**Example output:**



**Usage:** python All\_runs\_popn.py --save --version --datatype --xAxis --grey --runs --run\_file

**Example:** python All\_runs\_popn.py --save here --version v17p8 --datatype liq --xAxis pop --grey on --runs AlpineF2K Ashley JorKekNeed Cust --run\_file /home/fordw/Scripts/CCDF\_scripts\_and\_plots/Run\_test.txt

**Optional arguments:**

* **-s, --save:** Use “--save here” to save the figure to the current working directory. By default, the figure will save to /v17p8/Impact/ or to /v17p9/Impact/ depending on the version being run.
* **-v, --version:** Add “-v v17p9” to use runs from v17p9 or add “-v v17p8” to use runs from v17p8. By default, the script will use runs from v17p8.
* **-d, --datatype:** Use this to specify whether the areal probability data to plot is liquefaction or landslide. If the argument isn’t given, then the script will default to use liquefaction data. Recognised arguments: “--datatype liq”, “--datatype liquefaction”, “--datatype ls”, and “--datatype landslide”
* **-x, --xAxis:** Use this to change what is plotted on the x-axis. “--xAxis pop” will plot population on the x-axis, “--xAxis dwell” will plot dwellings on the x-axis, and leaving it out will plot area on the x-axis.   
  **Note:** population and dwelling data is currently unreliable (see popn\_db.txt and dwelling\_db.txt documentation)
* **-g, --grey:** Add “-g on” to plot all runs in grey. If any runs are specified with the --runs or --run\_file arguments, then these will be plotted in colour while all other runs will be in grey.
* **-r, --runs:** Use this argument to manually enter the runs to plot. If “--grey on” is used, then the manually entered runs will be plotted in colour whereas all other runs within v17p8 or v17p9 will be plotted in grey. Can be used in conjunction with the --run\_file argument.  
  **Example:** “--runs AlpineF2K AlpineK2T” will plot these two runs in colour
* **-f, --run\_file:** Use this argument to add a file of runs to plot. The file must have one run on each line. Works very similarly to --runs. Can work in conjunction with the --runs argument.

**Notes:**

* A maximum number of runs can be plotted in colour before the legend because too large for the figure. When too many runs are being plotted in colour, a message will appear explaining that all runs will be plotted in grey.
* If no runs are specified with the --runs or --run\_file arguments, then all runs within v17p8 or v17p9 will be plotted by default.
* The population and dwelling data isn’t the same between realisations making it difficult to calculate the mean of all realisations. To work around this, when population or dwellings are plotted on the x-axis, the script plots the first realisation of each run instead of the mean of all the realisations.
* There are two text files required for full functionality of the script:  
   popn\_db.txt  
   dwelling\_db.txt
* This script can often take several minutes to run (up to 30 minutes during testing)
* Since v17p9 has runs which have no data, the script cannot currently plot all v17p9 runs. It can still plot specific runs in v17p9 given they have the necessary data.
* The script is currently saved to /home/fordw/GroundFailure/scripts/

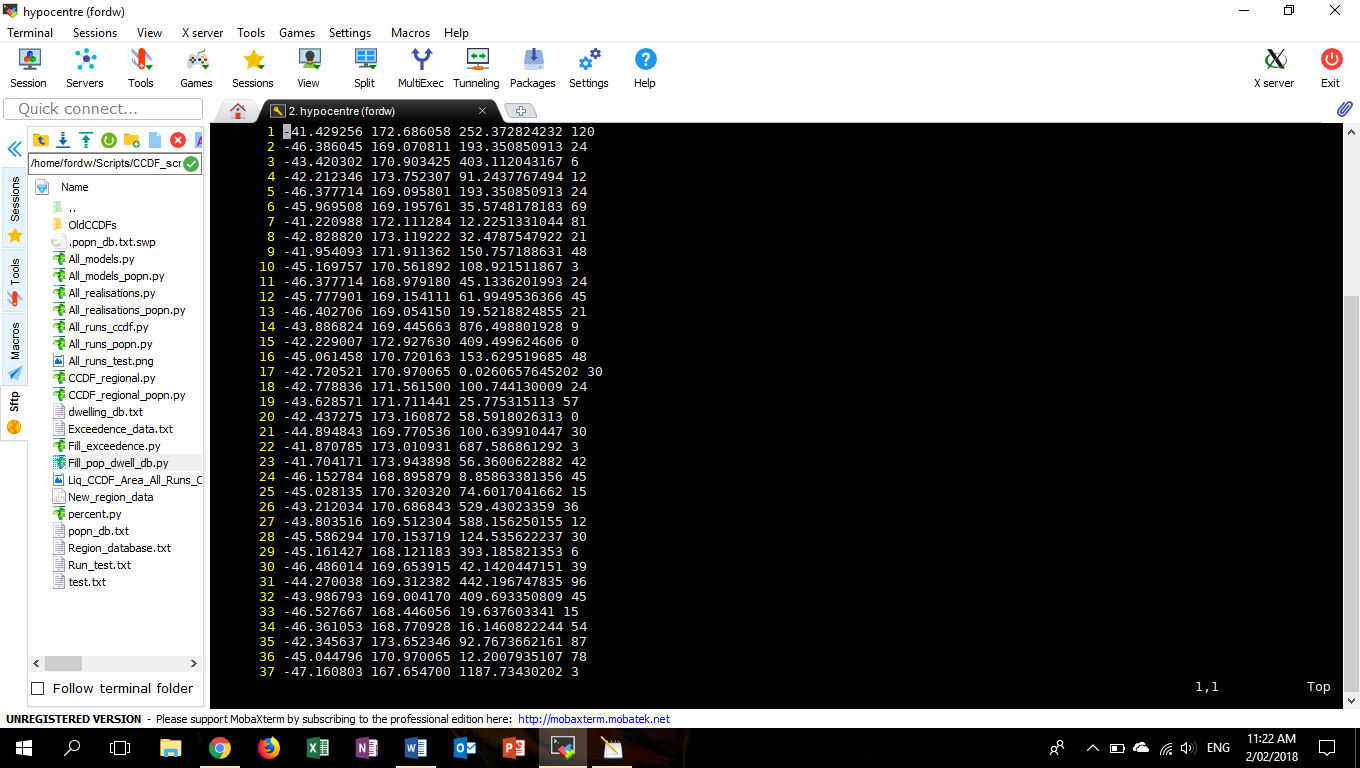
Next steps:

* The population and dwelling data needs to be validated (currently I do not believe it is reliable)
* In future, reference lines for population and dwellings could be added
* Could investigate a way to plot a mean line for the population/dwelling data

**Popn\_db.txt and dwelling\_db.txt**

The popn\_db.txt holds space separated latitude, longitude, area, population data whereas the dwelling\_db.txt holds space separated latitude, longitude, area, dwellings data on each line. The area and population/dwelling data is from the 2013 Census mesh block data. All data within these databases is added by using the Fill\_pop\_dwell\_db.py script (see documentation for further details)

**Example data:**



**Notes:**

* This data is required for plotting population or dwellings on the x-axis of any CCDF. The getPopOrDwell function within the CCDF scripts reads these files and puts the data into a useable dictionary
* The latitude and longitude values are from xyz files on Hypocentre. The Fill\_pop\_dwell\_db.py script finds what mesh block the latitude longitude point is in, and the area, pop/dwellings value is that of the entire mesh block. The CCDF scripts scale these values based on the size of the grid cells.
* Files are very large (400+ MB) and take ≈30 seconds to extract all data from.
* Currently saved to /home/fordw/Scripts/CCDF\_scripts\_and\_plots/

**Important:**

* There are many cases where a latitude/longitude gives a none value for area or population/dwellings. These inconsistencies reduce the reliability of the data when it is used within CCDFs. I recommend looking through the Fill\_pop\_dwell\_db.py script to find how to improve the data.

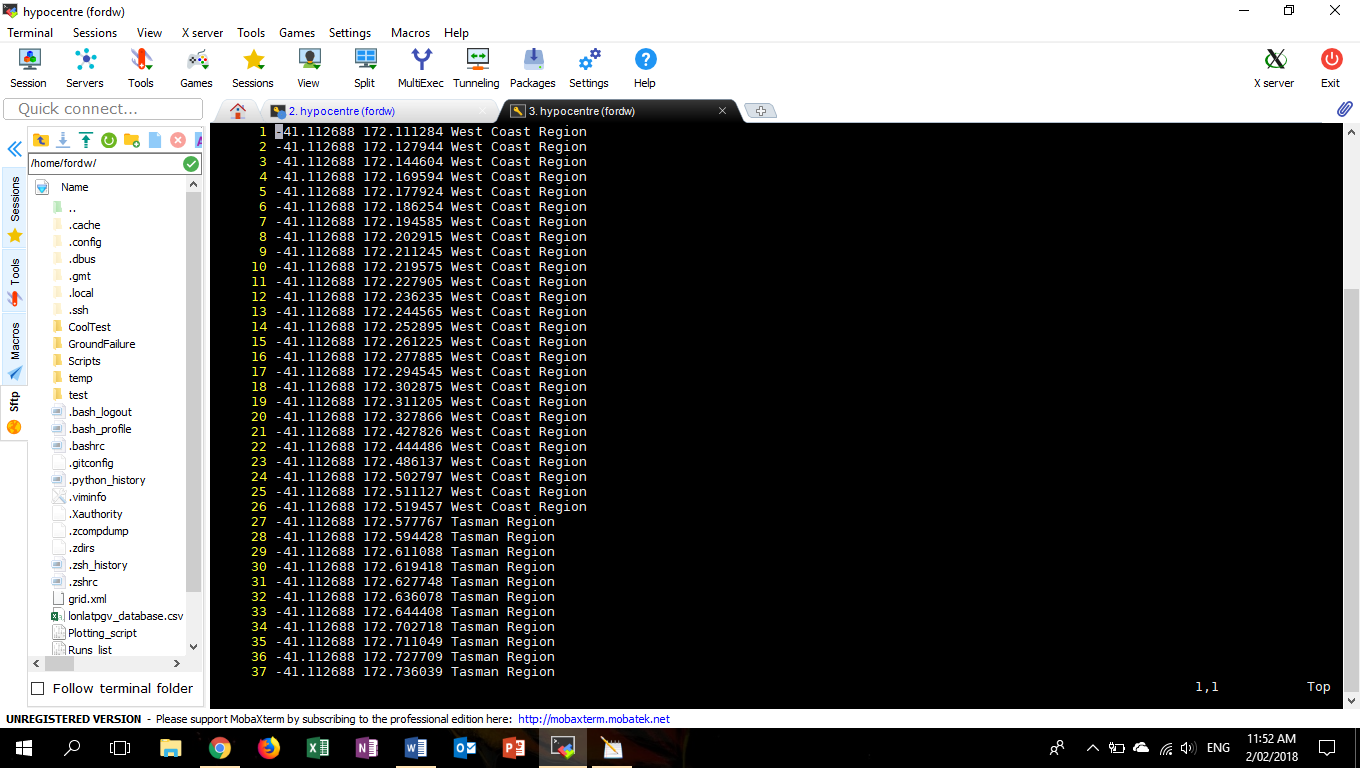
**Next steps:**

* When the 2018 Census data is available, these files will need to be replaced with updated versions
* Before producing a population/dwelling CCDF for a new run, make sure that the latitude/longitude data is added to the database. (use Fill\_pop\_dwell\_db.py)

**Region\_database.txt**

This file holds space separated latitude, longitude, region data on each line. The latitude/longitudes are from xyz files on Hypocentre. The region data is the region that the latitude longitude point is within (based off the region boundary data on koordinates.com). The file is required to plot regional CCDF subplots. The CCDF scripts will automatically update this file when they encounter data which is not already in the file.

**Example data:**



**Notes:**

* Updating this file can take several days because each new point must be checked with online data from koordinates.com. Once the data is in the file it is very fast to use within the CCDF scripts
* The getRegionData and updateRegionData functions (found in most CCDF scripts) read the data from this file and update it.
* Be careful when updating this file as latitude/longitude points which are not within a region of NZ are still added to the database and can cause problems.
* This file is relatively large (50+ MB) and adds around 6 seconds to CCDF script run time.
* If one user is updating the database, then another user trying to use the database may encounter error messages due to half filled lines in the database.
* Do not have multiple users trying to update at once - may corrupt the database.
* When updating this script will create a file called “New\_region\_data” which can be deleted at any stage after the script has run
* File is currently saved to /home/fordw/GroundFailure/scripts/

**Next steps:**

* Add a check to the updateRegionData function so that it does not add data when it has no corresponding region.
* When adding in lots of new data I recommend starting the update of this file so that the data is ready when it is needed.