**computeVs(z, qc, fs, u2, correlationName, correlationFlag):**

* Call **getCPTdata** to extract and trim the CPT measures
* Call **getCPTparam** to get basic parameters such as Ic, Qtn, etc from CPT measures
* From **getVsCorrelations** using correlation to estimate Vs profile
* Set the first 1.5m to a constant (set limits for shallow depth Vs, can’t be 0 or too large)
* If correlationFlag is **0**:
* Randomly generate 50 samples for each Vs
* If correlationFlag is **1**:
* Randomly generate 50 samples for the first row of Vs
* Calculate residual StandDev and lognormal StandDev
* Apply the same StandDev to the rest Vs, so it has a perfect correlation
* If correlationFlag is ‘**partial’**:
  + Call **computePartialCor**
* (optional) plot predicted Vs against measured Vs

Return: z, randVs, Vs

**computeVsz(z, qc, fs, u2, correlationName, correlationFlag):**

Calculate Vsz from randomly generated Vs profiles.

* Call **computeVs** to get z and an array of 50 randomly generated Vs profiles
* The mean Vsz is computed from Vs based on correlation
* The standard deviation of Vsz is estimated based on random realisation of Vs

Return: z, Vsz, Vsz\_sd

**computeVs30(filename, correlationName, correlationFlag):**

Calculate Vs30 from Vsz (Boore et al., 2011). This correlation does not consider the site class.

* Call **computeVsz** function to get z, Vsz, and Vsz\_sd
* Get coefficients [C0; C1; C2; σ] from table (z between 5 - 29 meters)
* Compute Vs30
* Compute standard deviation for Vs30 (first order second moment method):
* Note ‘ln(Vsz)’ is used in the derivative instead of ‘Vsz’ because the standard deviation is lognormal

Return: z, Vs30, Vs30\_SD

**CPT\_Vs30\_Main():**

* Inputs:
* **correlationNames** (‘McGann’, ‘Andrus’, ‘Hegazy’, ‘Robertson’, ‘McGann2)
* **correlationFlag** (0, 1, ‘partial’)
* It extracts CPT filenames and termination depth from a pre-existing file
* Loop through each CPT file using **computeVs30** function
* Write a **summary file** that contains: {ID}, {longitude}, {latitude}, {termDepth}, {Vs30}, {StandDev}

**getCPTparam(z, qc, fs, u2):**

* Assume area ratio = 0.8, soil weight = 1.9 (MN/m3), ground water table = 1 (m)
* Ic is calculated based on the non-normalised CPT soil behaviour type (Robertson, 2010).

You can also use iteration method to find normalised Ic, n and Qtn, results are pretty similar.

* qc1n and qt1n are assumed to be the same as Qtn, it doesn’t make a significant difference

Return: qt, Ic, Qtn, qc1n, qt1n, effective stress

**getCPTdata(filename):**

* Get CPT data from file
* Delete all rows that contain zero qc and fs
* Adjust the units

Return: z, qc, fs, u2

**computePartialCor(z, Vs, Vs\_SD, Nsim):**

* Please refer to the explanations in the script

Return: randVs, sigma

**getVsCorrelations:**

* **Vs\_McGann (z, qc, fs):**
* **kPa**

StandDev = 0.162 (z <= 5)

StandDev = 0.216 – 0.0108z (5 < z < 10)

StandDev = 0.108 (z > 10)

* **Vs\_McGann2(z, qc, fs):**
* **kPa**

StandDev = 0.2367

* **Vs\_Andrus (Ic, z, qt):**
* **kPa**

Holocene-Age Soils, where ASF = 1

residual standard deviation: suggests that 68% of the data fall within 24m/s.

* **Vs\_Robertson (z, Ic, Qtn, effectiveStress):**
* **MPa**

Since the standard deviation is not available in the paper, assume to be 0.2

* **Vs\_Hegazy (z, Ic, qc1n, effectiveStress):**
* **MPa**

Since the standard deviation is not available in the paper, assume to be 0.2