**Q1. What native features of Python support parallel programming?**

According to the Python 3.6 documentation, python supports parallel programming using these features that are present in the standard python package: threading, multiprocessing, concurrent sub-package and its futures module, subprocesses, schedulers and queues.

**Q2. What native features of R support parallel programming?**

R does not support parallel programming through any of its native features.

**Q3. What python libraries support parallel programming?**

The python wiki on parallel computing indicates that these packages and libraries support parallel programming in Python: dispy, delegate, pp, forkmap, process, processing, pyCSP, PyMP, remoteD, VecPy, batchlib, Celery, Deap, disco, DistributedPython, exec\_proxy, execnet, Ipython, job\_stream, jug, mpi4py, NetWorkSpaces, PaPy, papyrus pp, PyLinda, pyMPI, pypar, pyPastSet, SCOOP, seppo, superpy, SceintificPython package and its subpackages (Scientific.DistributedComputing.MasterSlave, Scientific.BSP, Scientific.MPI), and Star-P for Python.

The full list can be found at <https://wiki.python.org/moin/ParallelProcessing>

**Q4. What R libraries (packages) support parallel programming?**

The MGCV package and the parallel pacakage both support parallel programming in R. The MGCV package includes specialized modules like big additive model (BAM), general additive model (GAM), bindenv (useful for specifying locking procedures), and magic (enables the use of multiple cores or clusters). The full details on these packages and modules is available in the R reference index for R version 3.3.2

**Q5. Can GPUs be used with Python and R? If yes, then how?**

GPUs can be used with Python. As of now GPUs can only be used through the PyCUDA package in python. This package allows python to use the CUDA (Computer Unified Device Architecture) application programming interface created by Nvidia.

Meanwhile in R, GPUs can be used through the GPUr, gputools, permgpu, Cudabayesreg, nmfgpu4r, gcbd, and gmatrix packages. These packages can be found on the the list of available packages at <https://cran.r-project.org/>