Notes on integrating MultiPlotter with PolyPlotter

Testing:

* not working, even before Plotter changes:
  + tfTestJerome\_Larnder.py
  + tfTestJerome\_norot\_localopt.py
    - “tensorflow has no attribute “placeholder” “
  + tfTestJerome\_newsim\_noRot.py
    - “tensorflow has no attribute “placeholder” “
  + Test\_Curvature.py
* working before Plotter changes, now also working after:
  + - Test\_Plotter.py
    - Test\_CostFunction.py
    - TestLoadPlot.py; Test\_SpikePlot.py
    - Test\_Larnder.py
* TODO: collapse the following redundant tests into one file with a few variation functions to test:
  + Test\_Plotter.py, TestLoadPlot.py
  + ( I already deleted Test\_LoadOmega.py as it was redundant )

Interface and implementation changes:

* \*\* Seek volunteer: setTitle() not implemented:
* \*\* see volunteer: THings not working when using displayPoly() :
  + setCaptionText()
    - did not check if setCaptionValues() works
  + xAxisLabel not displaying
  + yAxisLabel not implemented ( part of displayPoly() arguments )
* appendCaptionValues() is not used directly in any tests: add it to the tests
  + value\_dict “default” ( containing the caption values ) removed from Plotter.py
    - should reside only in test code
* removed function call AccelData\_Rotate() from the test code: not relevant to the tests at hand
* test code for function Plot() was sitting in Plotter.py . Now moved to Test\_Plot.py
  + Plot() code written to explicitly handle only single RotaryData and AccelData objects.
    - Easier for newbies to follow how to use Plotter.py interface
* Test\_CostFunction.py
  + was not using displayPoly()
    - was hand-coding it because of custom plt.xlim, plt.ylim
    - now using DisplayPoly()
* applyStyle() only used internally, so renamed internal\_ApplyStyle()
* module/Load.py
  + renamed LoadRun() to LoadMultiRun() ( “LoadRun” available, but deprecated )
  + implemented LoadSingleRun()
  + naming scheme helps clarify things