import vertica\_sdk

class pyPartitions(vertica\_sdk.TransformFunction):

def processPartition(self, server\_interface, input, output):

rc = 0

rtotal = 0

asig = []

tsig = []

while True:

rc = rc + 1

rtotal = rtotal + input.getFloat(1)

tsig.append(input.getFloat(0))

asig.append(input.getFloat(1))

if not input.next():

break

server\_interface.log("pyPartition returning:"+str(rc)+","+str(rtotal/rc))

output.setFloat(0, rc)

output.setFloat(1, rtotal / rc)

output.next()

class pyPartitionsFactory(vertica\_sdk.TransformFunctionFactory):

def getPrototype(self, server\_interface, arg\_types, return\_type):

arg\_types.addFloat()

arg\_types.addFloat()

return\_type.addFloat()

return\_type.addFloat()

def getReturnType(self, server\_interface, arg\_types, return\_type):

return\_type.addColumn(arg\_types.getColumnType(0), "count")

return\_type.addColumn(arg\_types.getColumnType(1), "average")

def createTransformFunction(cls, server\_interface):

return pyPartitions()