package com.twitter.simclusters\_v2.scalding.common

import com.twitter.algebird.\_

/\*\*

\* The reason of creating this class is that we need multiple percentiles and current

\* implementations need one QTree per percentile which is unnecessary. This class gets multiple

\* percentiles from the same QTree.

\*/

case class QTreeMultiAggregator[T](percentiles: Seq[Double])(implicit val num: Numeric[T])

extends Aggregator[T, QTree[Unit], Map[String, Double]]

with QTreeAggregatorLike[T] {

require(

percentiles.forall(p => p >= 0.0 && p <= 1.0),

"The given percentile must be of the form 0 <= p <= 1.0"

)

override def percentile: Double = 0.0 // Useless but needed for the base class

override def k: Int = QTreeAggregator.DefaultK

private def getPercentile(qt: QTree[Unit], p: Double): Double = {

val (lower, upper) = qt.quantileBounds(p)

(lower + upper) / 2

}

def present(qt: QTree[Unit]): Map[String, Double] =

percentiles.map { p => p.toString -> getPercentile(qt, p) }.toMap

}