#Sys.setenv(HADOOP\_CMD="/usr/bin/hadoop")

#Sys.setenv(HADOOP\_STREAMING="/usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming.jar")

# puts the data into HDFS

# It is not possible to write out big data with to.dfs,

# to.dfs is nonetheless very useful for a variety of uses like writing test cases,

small.ints <- to.dfs(1:10)

# Version 1

from.dfs(

mapreduce(

input=small.ints,

map=function(k,v){

keyval(v,v^2)

},

reduce=function(k,v2){

keyval(k,v2)

}

)

)

# Version 2

from.dfs(

mapreduce(

input=small.ints,

map=function(k,v){

keyval(v,v^2)

}

)

)

# Sum

rmr.options.set(backend="local")

from.dfs(

mapreduce(

input=small.ints,

map=function(k,v){

keyval(1,sum(v))

},

reduce=function(k,v){

keyval(NULL,sum(v))

}

)

)

# Variance

from.dfs(

mapreduce(

input=small.ints,

map=function(k,v){

v=(v-5.5)^2

keyval(1,v)

},

reduce=function(k,v){

df=length(v)-1

keyval(NULL,sum(v)/df)

}

)

)

########################################################################################

groups<-rbinom(32,n=50,prob=0.4)

tapply(groups,groups,length)

groups<-to.dfs(groups)

from.dfs(

mapreduce(

input=groups,

map=function(.,v){

keyval(v,1)

},

reduce=function(k,vv){

keyval(k,length(vv))

}

)

)