참고자료

2017년 10월 22일 일요일 오후 7:02

```
install.packages("gplots")
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-
streaming-2.8.1.jar")
library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)
library(dplyr)
library(reshape2)
library(gplots)
hdfs.del("/airdata/ex1")

inputfile <- "/airdata/input/2008_sub.csv"
#### map function
mapper_flight = function(.,fields) {
  delay = as.numeric(as.character(fields[[16]]))
  filter = !is.na(delay)
  year = as.character(fields[[1]])
  month = as.character(fields[[2]])
  day = as.character(fields[[3]])
  carrier = as.character(fields[[9]])
  key.df = data.frame(
    date = paste(
      year[filter],
      month[filter],
      day[filter],
      sep='-'),
    carrier = carrier[filter] )
  output.val = data.frame( delay = delay[filter] )
  output.val$delay = as.numeric(output.val$delay)
  keyval( key.df , output.val )
}
### reduce function
reducer = function(k, v) {
  output.val = data.frame(
    avg = mean(v$delay, na.rm=T))
  keyval(k, output.val)
}
### mapreduce
result = mapreduce(
  input = inputfile,
  output = "/tmp/ex1",
  input.format = make.input.format("csv", sep = ","),
  map = mapper_flight,
  reduce = reducer )
###### preprocessing
data = from.dfs(result)
data = as.data.frame(data, stringsAsFactors = F)
colnames(data) = c("date", "carrier", "avg")
```

```
data = transform(
  data ,
  date = as.Date(date) ,
  carrier = as.character(carrier) ,
  avg = scale(avg)
)
mat_carrierByDate = dcast(data, carrier ~ date)
row.names(mat_carrierByDate) = mat_carrierByDate$carrier
mat_carrierByDate = mat_carrier
ByDate[,-1]
mat_carrierByDate = data.matrix(mat_carrierByDate)

date = colnames(mat_carrierByDate)
month = seq(from=1, to = length(date), by = 30)
date[-month] = ""
##### visualization
res_matrix = data.matrix(mat_carrierByDate)
heatmap.2(res_matrix,dendrogram="none",trace="none",scale="column", Rowv=FALSE,
Colv=FALSE, main = paste0("year:",2008), key = T, density.info="none", symkey=FALSE, ,cexRow=
1,cexCol=1, margins=c(5,5), labCol = date, srtCol = 45)
```

총 운항횟수 구하기

2017년 10월 22일 일요일 오후 11:15

```
#rhdfs에서 활용하기 위한 환경 변구
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
#mapreduce에서 스트리밍 형식으로 데이터를 읽어 들이기 때문에 라이브러리 등록
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar")
#rhdfs에서 활용하기 위해 등록
library(rJava)
#hadoop과 연동하기 위해 사용
library(rhdfs)
hdfs.init()
#mapreduce를 사용하기 위해 등록
library(rmr2)

#입력 데이터 및 출력 위치 지정
inputFile<-"/airData/2008_sub.csv"
outputDir<-"/airData/Ex01"

#mapper
mapper=function(.,dataArr){
 #dataArr[[1]]값을 character로 변경해서 year에 저장
 year <- as.character(dataArr[[1]])
 #reducer로 전달
 keyval(year, 1)
}
#reducer

reducer=function(k,v){
 keyval(k, sum(v))
}

#결과를 저장한 디렉토리로 이미 생성되어 있다면 삭제
hdfs.del("/airData/Ex01")
#main
result=mapreduce(
 input = inputFile,
 output = outputDir,
 input.format = make.input.format("csv", sep=","),
 map = mapper,
 reduce = reducer
)
#실행 결과 확인
hdfs.ls("/airData/Ex01")
result

#from.dfs는 하둡에서 결과를 얻어오는 함수로 실행 결과를 읽어 들인다.
```

#/airData/Ex01/part-00000을 읽어 들이는 것이다.

```
fileData <- from.dfs(result)
fileData
fileData$val
```

월별 운항 횟수 구하기

2017년 10월 23일 월요일 오전 9:21

```
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-
streaming-2.8.1.jar")

library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)

hdfs.del("/airdata/ex2")
inputfile<-"/airdata/input/2008_sub.csv"

mapper<-function(.,dataArr){
  year = as.character(dataArr[[1]])
  month = as.character((dataArr[[2]]))
  keyData = paste(year,"-",month)
  keyval(keyData, 1)
}
reducer<-function(key, value){
  keyval(key, length(value))
}

result<-mapreduce(
  input=inputfile,
  output="/airdata/ex2",
  input.format = make.input.format("csv", sep=","),
  map=mapper,
  reduce = reducer
)

plot(fileData$val, type="l")
```

요일별 운항 정보

2017년 10월 23일 월요일 오후 2:40

월요일
화요일
수요일
목요일
금요일
토요일
일요일

```
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar")

library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)

hdfs.del("/airData/ex3")
inputfile<-"/airData/2008_sub.csv"

mapper<-function(.,dataArr){
  year = as.character(dataArr[[1]])
  month = as.character(dataArr[[2]])
  weekInt = as.integer(dataArr[[4]])

##############################################################################################
  # week='월요일'
  # if(weekInt==2)    week='화요일'
  # else if(weekInt==3)    week='수요일'
  # else if(weekInt==4)    week='목요일'
  # else if(weekInt==5)    week='금요일'
  # else if(weekInt==6)    week='토요일'
  # else if(weekInt==7)    week='일요일'
##############################################################################################

week<-c('월요일', '화요일', '수요일', '목요일',  '금요일', '토요일', '일요일')

  keyData = paste(year,"-",month, "-", week[weekInt])
  keyval(keyData, 1)
}
reducer<-function(key, value){
  keyval(key, sum(value))
}

result<-mapreduce(
```

```
  input=inputfile,
  output="/airdata/ex3",
  input.format = make.input.format("csv", sep=","),
  map=mapper,
  reduce = reducer
)
fileData=from.dfs(result)
fileData

plot(fileData$val, type="h")
```

월별 결항 횟수

2017년 10월 26일 목요일 오전 10:34

```
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-
streaming-2.8.1.jar")

library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)

inputfile<-"/airData/2008_sub.csv"
outputDir<-"/airData/ex4"

mapper<-function(.,dataArr){
 year = as.character(dataArr[[1]])
 month = as.character(dataArr[[2]])
 weekInt = as.integer(dataArr[[4]])
 depTime = as.numeric(as.character(dataArr[[5]]))
 filter = is.na(depTime)

 keyData<-paste("NA",year[filter],month[filter], sep = "-")

 keyval(keyData, 1)
}
reducer<-function(key, value){
 if(is.na(match(key, "NA--"))){
  key<-sub("NA-", "", key)
  keyval(key, sum(value))
 }
}

hdfs.del(outputDir)
result<-mapreduce(
 input=inputfile,
 output=outputDir,
 input.format = make.input.format("csv", sep=","),
 map=mapper,
 reduce = reducer,
 combine = T
)
fileData=from.dfs(result)
fileData

plot(fileData$val, type="h")
```

월별 실출항 횟수 구하기

2017년 10월 25일 수요일 오후 2:02

```
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-
streaming-2.8.1.jar")

library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)

inputfile<-"/airData/2008_sub.csv"
outputDir<-"/airData/ex4"

mapper<-function(.,dataArr){
  year = as.character(dataArr[[1]])
  month = as.character(dataArr[[2]])
  weekInt = as.integer(dataArr[[4]])
  depTime = as.numeric(as.character(dataArr[[5]]))
  filter = !is.na(depTime)

  keyData<-paste(year[filter],month[filter], sep = "-")
  keyval(keyData, 1)
}
reducer<-function(key, value){
  keyval(key, sum(value))
}

hdfs.del(outputDir)
result<-mapreduce(
  input=inputfile,
  output=outputDir,
  input.format = make.input.format("csv", sep=","),
  map=mapper,
  reduce = reducer,
  combine = T
)
fileData=from.dfs(result)
fileData

plot(fileData$val, type="h")
```

월별 운항 횟수 & 실운항 횟수

2017년 10월 26일 목요일 오전 10:30

```
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-
streaming-2.8.1.jar")

library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)

inputfile<-"/airData/2008_sub.csv"
outputDir<-"/airData/ex4"

mapper<-function(.,dataArr){
 year = as.character(dataArr[[1]])
 month = as.character(dataArr[[2]])
 weekInt = as.integer(dataArr[[4]])
 depTime = as.numeric(as.character(dataArr[[5]]))
 filter = !is.na(depTime)

 keyData<-c(
  paste("real",year[filter],month[filter], sep = "-"),
  paste("total",year,month, sep = "-"))

 keyval(keyData, 1)
}
reducer<-function(key, value){
 keyval(key, sum(value))
}

hdfs.del(outputDir)
result<-mapreduce(
 input=inputfile,
 output=outputDir,
 input.format = make.input.format("csv", sep=","),
 map=mapper,
 reduce = reducer,
 combine = T
)
fileData=from.dfs(result)
fileData

plot(fileData$val, type="h")
```

일자별 운항 횟수

2017년 10월 26일 목요일 오전 11:35

```
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-
streaming-2.8.1.jar")

library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)

inputfile<-"/airData/2008_sub.csv"
outputDir<-"/airData/ex4"

getDay<-function(dataArr){
  day = as.numeric(dataArr[[3]])
  day = (day-1)%/%10#1~10, 21
  day = day-(day%/%3)#0, 12, 3
  return (day)
}

mapper<-function(.,dataArr){
  year = as.character(dataArr[[1]])
  month = as.character(dataArr[[2]])
  day = getDay(dataArr)
  keyData<-paste(year,month,day, sep = "-")

  keyval(keyData, 1)
}
#reducer에서 사용되는 함수로 2번 호출되어 조건문으로 처리함
#첫 번째 들어올 경우 조건문을 거치게 되고 두 번째 들어올 경우 if문 다음의 return문으로 이동
하게 됨. return이
#없을 경우 key값이 빈 값으로 나타남
convertKey<-function(str){
  rangeArr<-c("1~10", "11~20", "21~31")
  len<-nchar(str)
  pos<-substr(str, len-1, len)
  if(substr(pos, 1, 1)=='-'){
    pos=as.numeric(substr(pos,2,2))
    headStr<-substr(str, 1, len-2)
    return (paste(headStr, rangeArr[pos+1], sep = " : "))
  }
  return (str)
}

reducer<-function(key, value){
  key<-convertKey(key)
  keyval(key, sum(value))
}
```

```
hdfs.del(outputDir)
result<-mapreduce(
  input=inputfile,
  output=outputDir,
  input.format = make.input.format("csv", sep=","),
  map=mapper,
  reduce = reducer,
  combine = T
)
fileData=from.dfs(result)
fileData

plot(fileData$val, type="l")
```

항공사별 도착 지연 횟수

2017년 10월 26일 목요일 오후 3:08

```
Sys.setenv(HADOOP_CMD="/usr/local/hadoop/bin/hadoop")
Sys.setenv(HADOOP_STREAMING="/usr/local/hadoop/share/hadoop/tools/lib/hadoop-
streaming-2.8.1.jar")

library(rJava)
library(rhdfs)
hdfs.init()
library(rmr2)

inputfile<-"/airData/2008_sub.csv"
outputDir<-"/airData/ex4"

getArrDelay<-function(dataArr){
  uniqueCarrier = as.character(dataArr[[9]])
  arrDelay<-as.numeric(as.character(dataArr[[15]]))
  filter<-!is.na(arrDelay)
  valNum<-ifelse(arrDelay[filter]>0,yes = 1,no = 0)
  retVal_frame<-data.frame(key=uniqueCarrier[filter],
                           val=valNum)
  return (retVal_frame)
}
mapper<-function(.,dataArr){
  keyvalue<-getArrDelay(dataArr)

  keyval(keyvalue$key, keyvalue$val)
}

reducer<-function(key, value){
  keyval(key, sum(value))
}

hdfs.del(outputDir)
result<-mapreduce(
  input=inputfile,
  output=outputDir,
  input.format = make.input.format("csv", sep=","),
  map=mapper,
  reduce = reducer,
  combine = T
)
fileData=from.dfs(result)
fileData

plot(fileData$val, type="l")
```