Homework1.R

student

Wed Feb 3 12:10:56 2016

##CS 480 Homework 1  
##Neeraj Asthana (nasthan2)  
  
library(RSQLite)

## Loading required package: DBI

library(biganalytics)

## Loading required package: bigmemory

## Loading required package: bigmemory.sri

## Loading required package: foreach

## Loading required package: biglm

library(foreach)  
  
#Create database connection  
setwd("/home/student/container-data/RDataScience/AirlineDelays")  
delay.con <- dbConnect(RSQLite::SQLite(), dbname = "AirlineDelay.sqlite3")  
  
  
##Q3  
#flights per day of the week  
dbGetQuery(delay.con,   
 "SELECT DayOfWeek, COUNT(DayOfWeek) FROM AirlineDelay WHERE DayOfWeek <> 0 GROUP BY DayOfWeek")

## DayofWeek COUNT(DayOfWeek)  
## 1 1 18136111  
## 2 2 18061938  
## 3 3 18103222  
## 4 4 18083800  
## 5 5 18091338  
## 6 6 15915382  
## 7 7 17143178  
## 8 0 1

##Q4  
#flights per day of the week for each week   
dbGetQuery(delay.con, "SELECT Year,DayOfWeek, COUNT(DayOfWeek) FROM AirlineDelay WHERE DayOfWeek <> 0 GROUP BY Year,DayOfWeek")

## Year DayofWeek COUNT(DayOfWeek)  
## 1 1987 1 190711  
## 2 1987 2 190238  
## 3 1987 3 190235  
## 4 1987 4 200911  
## 5 1987 5 184913  
## 6 1987 6 173370  
## 7 1987 7 181448  
## 8 1988 1 755898  
## 9 1988 2 757140  
## 10 1988 3 757963  
## 11 1988 4 753415  
## 12 1988 5 766364  
## 13 1988 6 697795  
## 14 1988 7 713521  
## 15 1989 1 733459  
## 16 1989 2 735404  
## 17 1989 3 735180  
## 18 1989 4 731008  
## 19 1989 5 731548  
## 20 1989 6 668080  
## 21 1989 7 706521  
## 22 1990 1 781823  
## 23 1990 2 770341  
## 24 1990 3 770044  
## 25 1990 4 766575  
## 26 1990 5 765068  
## 27 1990 6 694251  
## 28 1990 7 722791  
## 29 1991 1 747531  
## 30 1991 2 758840  
## 31 1991 3 745401  
## 32 1991 4 739321  
## 33 1991 5 739313  
## 34 1991 6 652991  
## 35 1991 7 693528  
## 36 1992 1 745442  
## 37 1992 2 746435  
## 38 1992 3 758213  
## 39 1992 4 751417  
## 40 1992 5 736609  
## 41 1992 6 656920  
## 42 1992 7 697121  
## 43 1993 1 741084  
## 44 1993 2 743108  
## 45 1993 3 741988  
## 46 1993 4 737026  
## 47 1993 5 746019  
## 48 1993 6 663606  
## 49 1993 7 697670  
## 50 1994 1 754636  
## 51 1994 2 756532  
## 52 1994 3 756864  
## 53 1994 4 751537  
## 54 1994 5 751531  
## 55 1994 6 695245  
## 56 1994 7 713703  
## 57 1995 1 773467  
## 58 1995 2 778389  
## 59 1995 3 779452  
## 60 1995 4 775836  
## 61 1995 5 776992  
## 62 1995 6 695286  
## 63 1995 7 748013  
## 64 1996 1 793071  
## 65 1996 2 792543  
## 66 1996 3 778193  
## 67 1996 4 774533  
## 68 1996 5 775343  
## 69 1996 6 699527  
## 70 1996 7 738773  
## 71 1997 1 790298  
## 72 1997 2 791617  
## 73 1997 3 802130  
## 74 1997 4 785731  
## 75 1997 5 786342  
## 76 1997 6 706198  
## 77 1997 7 749527  
## 78 1998 1 788003  
## 79 1998 2 789241  
## 80 1998 3 789076  
## 81 1998 4 796404  
## 82 1998 5 782299  
## 83 1998 6 694528  
## 84 1998 7 745170  
## 85 1999 1 810519  
## 86 1999 2 812478  
## 87 1999 3 812857  
## 88 1999 4 809243  
## 89 1999 5 817145  
## 90 1999 6 704006  
## 91 1999 7 761636  
## 92 2000 1 825186  
## 93 2000 2 828523  
## 94 2000 3 830751  
## 95 2000 4 827740  
## 96 2000 5 828944  
## 97 2000 6 741662  
## 98 2000 7 800241  
## 99 2001 1 881379  
## 100 2001 2 871342  
## 101 2001 3 871772  
## 102 2001 4 870683  
## 103 2001 5 871140  
## 104 2001 6 772781  
## 105 2001 7 828683  
## 106 2002 1 774285  
## 107 2002 2 780556  
## 108 2002 3 769799  
## 109 2002 4 768973  
## 110 2002 5 771174  
## 111 2002 6 674222  
## 112 2002 7 732350  
## 113 2003 1 959114  
## 114 2003 2 947126  
## 115 2003 3 962100  
## 116 2003 4 952542  
## 117 2003 5 954701  
## 118 2003 6 811260  
## 119 2003 7 901697  
## 120 2004 1 1044508  
## 121 2004 2 1033863  
## 122 2004 3 1036521  
## 123 2004 4 1060245  
## 124 2004 5 1061447  
## 125 2004 6 903807  
## 126 2004 7 988879  
## 127 2005 1 1048968  
## 128 2005 2 1037289  
## 129 2005 3 1043208  
## 130 2005 4 1047749  
## 131 2005 5 1050985  
## 132 2005 6 919442  
## 133 2005 7 992955  
## 134 2006 1 1048054  
## 135 2006 2 1030322  
## 136 2006 3 1042952  
## 137 2006 4 1052949  
## 138 2006 5 1056606  
## 139 2006 6 899531  
## 140 2006 7 1011508  
## 141 2007 1 1112474  
## 142 2007 2 1078562  
## 143 2007 3 1088858  
## 144 2007 4 1097738  
## 145 2007 5 1101689  
## 146 2007 6 933338  
## 147 2007 7 1040556  
## 148 2008 1 1036201  
## 149 2008 2 1032049  
## 150 2008 3 1039665  
## 151 2008 4 1032224  
## 152 2008 5 1035166  
## 153 2008 6 857536  
## 154 2008 7 976887  
## 155 0 0 1

##Q5  
#TailNum  
dbGetQuery(delay.con, "SELECT Year, COUNT(\*) FROM AirlineDelay WHERE TailNum = 'NA' GROUP BY Year")

## Year COUNT(\*)  
## 1 1987 1311826  
## 2 1988 5202096  
## 3 1989 5041200  
## 4 1990 5270893  
## 5 1991 5076925  
## 6 1992 5092157  
## 7 1993 5070501  
## 8 1994 5180048

##Q7  
#minimize average delay by day of the week  
y <- attach.big.matrix("airline.desc")  
  
#Get mean departure delays for each day  
totalDayDelays <- foreach(i = 1:7, .combine=c) %do% {  
 mean(y[y[,"DayOfWeek"]==i, "DepDelay"], na.rm=TRUE)  
}  
  
#Therefore, Tuesday is the best day to minimize delays as it has the least average departure delay time  
totalDayDelays

## [1] 7.850057 6.855870 7.651197 9.246910 10.151539 6.887023 8.409293

#Get mean departure delays for each hour  
totalHourDelays <- foreach(i = 0:24, .combine=c) %do% {  
 mean(y[floor(y[,"CRSDepTime"]/100)==i, "DepDelay"], na.rm=TRUE)  
}  
#Midnight 0 and 24 are the same thing and must be treated as such  
#Take the mean of all CRSDepTimes that evaluate to 0 or 24 and store in the first index of totalHourDelays  
#I will not include index 25 in my final results (corresponds to only 24 value)  
midnight <- c(y[floor(y[,"CRSDepTime"]/100)==0, "DepDelay"], y[floor(y[,"CRSDepTime"]/100)==24, "DepDelay"])  
totalHourDelays[1] <- mean(midnight, na.rm=TRUE)  
  
#Therefore, 5:00AM is the best hour to minimize departure delays (1.532 min in delays)  
totalHourDelays[1:24]

## [1] 9.034998 5.249016 1.999306 2.826125 2.786590 1.531860 1.874374  
## [8] 2.840858 4.158435 4.972405 5.722319 6.524786 7.117775 7.940909  
## [15] 8.970806 9.992722 10.845703 12.232812 12.806944 13.645763 13.597240  
## [22] 12.324804 9.797672 8.331110

#disconnect from the database  
dbDisconnect(delay.con)

## [1] TRUE