BigDeals

Anubhav Dixit

2022-11-15

```
#BigDeal Challenge model development
# install.packages("ipred", repos="http://R-Forge.R-project.org")
# install.packages("fpp")
# install.packages("forecast")
# install.packages("ffp3")
# install.packages("tidyverse")
# install.packages("xlsx")

library("xlsx")
```

```
## Warning: package 'xlsx' was built under R version 4.2.2
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.2.2
```

```
library(readxl)
library(rpart)
library(ipred)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(npreg)
library(fpp)
```

```
## Warning: package 'fpp' was built under R version 4.2.2

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 4.2.2

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
## Loading required package: fma

## Warning: package 'fma' was built under R version 4.2.2

## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 4.2.2

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.2.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.2.2

##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: tseries

## Warning: package 'tseries' was built under R version 4.2.2
```

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.2.2

## Loaded gbm 2.1.8.1
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.2.2
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
Qualifying_Math_Data <- read_excel("C:/Users/Shashank/Desktop/INDE 6360/BigDEAL Challenge 2022 Qualifyi
df3<-data.frame(Qualifying_Math_Data)
```

```
rmse_reg <- function(model_obj, testing = NULL, target = NULL) {
  #Calculates rmse for a regression decision tree
  #Arguments:
  # testing - test data set
  # target  - target variable (length 1 character vector)
  yhat <- predict(model_obj, newdata = testing)
  actual <- testing[[target]]
  sqrt(mean((yhat-actual)^2))
}
```

```
#create a date column
#df3$Date<-as.Date(with(df3,paste(Year,Month,Day,sep="-")),"%Y-%m-%d")
#head(df3)

#subset the data for all years prior to 2007
#This approach processes all the data, spring summer and fall
df00<-subset(df3, subset = df3$Year< 2007)
head(df00)
```

```
##   Year Month Day Weekday Hour Tavg Tmed Tmax Tmin    Load
## 1 2002     1   1       3    1   43   43   60   31 1384494
## 2 2002     1   1       3    2   42   42   58   29 1392822
## 3 2002     1   1       3    3   41   41   57   31 1407887
## 4 2002     1   1       3    4   41   41   56   30 1438658
## 5 2002     1   1       3    5   40   41   53   29 1484046
## 6 2002     1   1       3    6   39   39   52   29 1559169
```

```
tail(df00)
```

```
##       Year Month Day Weekday Hour Tavg Tmed Tmax Tmin    Load
## 43819 2006    12  31       1   19   72   73   77   65 1706361
## 43820 2006    12  31       1   20   71   72   77   65 1612494
## 43821 2006    12  31       1   21   70   72   75   65 1473990
```

```
## 43822 2006    12  31       1   22   70   71   75   65 1374181
## 43823 2006    12  31       1   23   69   70   74   65 1272117
## 43824 2006    12  31       1   24   69   70   73   64 1165956
```

```
#Create training and test data sets
#make this example reproducible
set.seed(1)
rows <- sample(x=nrow(df00), size=.7*nrow(df00))
data.train <- df00[rows,]
data.test <- df00[-rows,]

#Spring load predictions with randomForest()
annual.fit <- randomForest(Load ~ ., data = data.train, mtry = 2, n.trees = 10000)
annual.error<-rmse_reg(annual.fit, data.test, "Load")
annual.error
```

```
## [1] 84300.03
```

```
summary(annual.fit)
```

```
##                 Length Class  Mode
## call               5  -none- call
## type               1  -none- character
## predicted      30676  -none- numeric
## mse              500  -none- numeric
## rsq              500  -none- numeric
## oob.times      30676  -none- numeric
## importance         9  -none- numeric
## importanceSD       0  -none- NULL
## localImportance    0  -none- NULL
## proximity          0  -none- NULL
## ntree              1  -none- numeric
## mtry               1  -none- numeric
## forest            11  -none- list
## coefs              0  -none- NULL
## y              30676  -none- numeric
## test               0  -none- NULL
## inbag              0  -none- NULL
## terms              3  terms  call
```

```
spring.train <- subset(data.train, subset = data.train$Month >= 1 & data.train$Month <= 4)
spring.test <- subset(data.test, subset = data.test$Month >= 1 & data.test$Month <= 4)
summer.train <- subset(data.train, subset = data.train$Month >= 5 & data.train$Month <= 10)
summer.test <- subset(data.test, subset = data.test$Month >= 5 & data.test$Month <= 10)
fall.train <- subset(data.train, subset = data.train$Month >= 11 & data.train$Month <= 12)
fall.test <- subset(data.test, subset = data.test$Month >= 11 & data.test$Month <= 12)

head(spring.train)
```

```
##       Year Month Day Weekday Hour Tavg Tmed Tmax Tmin    Load
## 11571 2003     4  28       2    3   64   64   71   58  667050
## 26954 2005     1  28       6    2   55   55   62   46  896430
```

```
## 36244 2006     2  19       1    4   56   58   62   45  776081
## 26663 2005     1  15       7   23   55   54   67   48 1401117
## 19242 2004     3  12       6   18   69   70   76   64 1116959
## 9392  2003     1  27       2    8   39   37   49   30 2308728
```

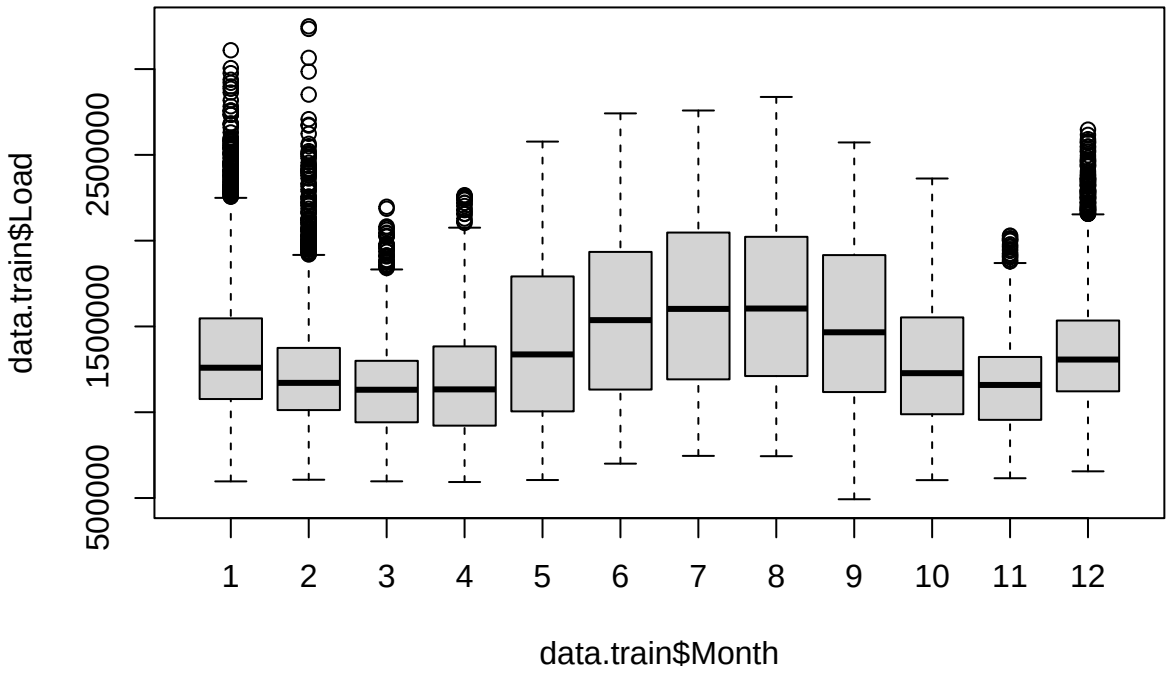
```
head(summer.train)
```

```
##       Year Month Day Weekday Hour Tavg Tmed Tmax Tmin    Load
## 24388 2004    10  13       4    4   70   71   75   64  826107
## 4050  2002     6  18       3   18   77   76   88   72 1393294
## 32618 2005     9  21       4    2   77   78   82   74 1245314
## 13903 2003     8   3       1    7   72   72   79   69  872073
## 22306 2004     7  18       1   10   78   77   86   74 1383551
## 39294 2006     6  26       2    6   73   73   77   71 1078136
```

```
head(fall.train)
```

```
##       Year Month Day Weekday Hour Tavg Tmed Tmax Tmin    Load
## 43307 2006    12  10       1   11   66   68   74   58 1527745
## 25173 2004    11  14       1   21   65   66   74   55 1291268
## 8229  2002    12   9       2   21   60   61   70   51 1370632
## 25305 2004    11  20       7    9   67   69   73   55 1059242
## 25061 2004    11  10       4    5   61   63   73   49  779586
## 43809 2006    12  31       1    9   70   70   76   64 1106358
```

```
#explore the load variation through the months
boxplot(data.train$Load~data.train$Month)
```




```
#Spring load predictions with randomForest()
spring.fit <- randomForest(Load ~ ., data = spring.train, mtry = 2, n.trees = 10000)
spring.error<-rmse_reg(spring.fit, spring.test, "Load")
spring.error
```

```
## [1] 90292.97
```

```
summary(spring.fit)
```

```
##                 Length Class  Mode
## call                 5 -none- call
## type                 1 -none- character
## predicted        10022 -none- numeric
## mse                500 -none- numeric
## rsq                500 -none- numeric
## oob.times        10022 -none- numeric
## importance           9 -none- numeric
## importanceSD         0 -none- NULL
## localImportance      0 -none- NULL
## proximity            0 -none- NULL
## ntree                1 -none- numeric
## mtry                 1 -none- numeric
## forest              11 -none- list
## coefs                0 -none- NULL
```

```
## y               10022  -none- numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
#Summer load predictions with randomForest()
summer.fit <- randomForest(Load ~ ., data = summer.train, mtry = 2, n.trees = 10000)
summer.error<-rmse_reg(summer.fit, summer.test, "Load")
summer.error
```

```
## [1] 78218.71
```

```
summary(summer.fit)
```

```
##                 Length Class  Mode
## call                5  -none- call
## type                1  -none- character
## predicted       15507  -none- numeric
## mse               500  -none- numeric
## rsq               500  -none- numeric
## oob.times       15507  -none- numeric
## importance          9  -none- numeric
## importanceSD        0  -none- NULL
## localImportance     0  -none- NULL
## proximity           0  -none- NULL
## ntree               1  -none- numeric
## mtry                1  -none- numeric
## forest             11  -none- list
## coefs               0  -none- NULL
## y               15507  -none- numeric
## test                0  -none- NULL
## inbag               0  -none- NULL
## terms               3  terms  call
```

```
#fall load predictions with randomForest()
fall.fit <- randomForest(Load ~ ., data = fall.train, mtry = 2, n.trees = 10000)
fall.error<-rmse_reg(fall.fit, fall.test, "Load")
fall.error
```

```
## [1] 92920.41
```

```
summary(fall.fit)
```

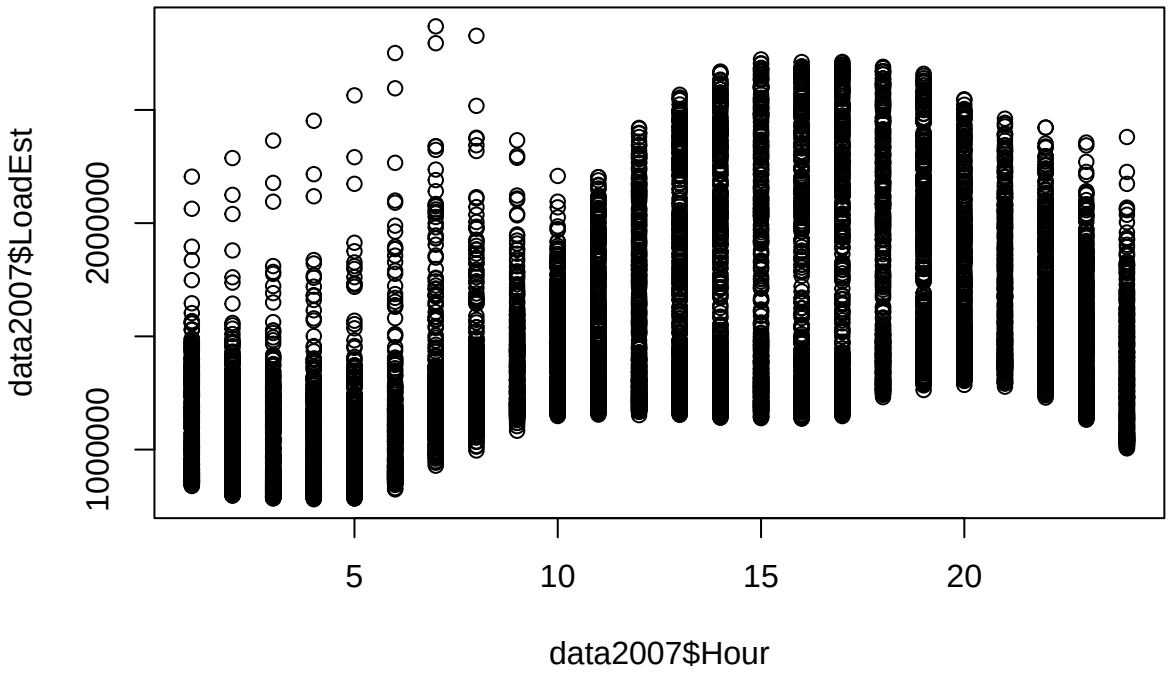
```
##                 Length Class  Mode
## call                5  -none- call
## type                1  -none- character
## predicted       5147  -none- numeric
## mse              500  -none- numeric
## rsq              500  -none- numeric
## oob.times       5147  -none- numeric
## importance         9  -none- numeric
```

```
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            11   -none- list
## coefs              0   -none- NULL
## y               5147   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```

```
#Prepare prediction vectors for 2007

data2007<-data.frame(subset(df3, subset = df3$Year== 2007))

spring2007<-data.frame(subset(data2007, subset = data2007$Month >= 1 & data2007$Month <= 4))
spring.forecast<-predict(spring.fit, newdata = spring2007)
write.table(spring.forecast, file = "springForecast.csv", sep = ",", col.names = NA)

summer2007<-data.frame(subset(data2007, data2007$Month >= 5 & data2007$Month <= 10))
summer.forecast<-predict(summer.fit, newdata = summer2007)
write.table(summer.forecast, file = "summerForecast.csv", sep = ",", col.names = NA)

fall2007<-data.frame(subset(data2007, subset = data2007$Month >= 11 & data2007$Month <= 12))
fall.forecast<-predict(fall.fit, newdata = fall2007)
write.table(fall.forecast, file = "fallForecast.csv", sep = ",", col.names = NA)

#Prepare Tracks 2 & 3
#begin working with annual data set

annual.forecast<-predict(annual.fit, newdata = data2007)
data2007$Date<-as.Date(with(data2007,paste(Year,Month,Day,sep="-")),"%Y-%m-%d")
data2007$LoadEst<-annual.forecast
head(data2007)
```

```
##       Year Month Day Weekday Hour Tavg Tmed Tmax Tmin Load       Date  LoadEst
## 43825 2007     1   1       2    1   69   70   73   64   NA 2007-01-01 924704.8
## 43826 2007     1   1       2    2   68   69   73   64   NA 2007-01-01 851004.5
## 43827 2007     1   1       2    3   68   68   73   64   NA 2007-01-01 841368.0
## 43828 2007     1   1       2    4   67   68   72   63   NA 2007-01-01 816486.5
## 43829 2007     1   1       2    5   67   68   72   63   NA 2007-01-01 816741.2
## 43830 2007     1   1       2    6   67   68   72   61   NA 2007-01-01 858538.0
```

```
plot(data2007$LoadEst~data2007$Hour)
```




```
  # dailyTemps<-subset(data2007, subset = data2007$Month == 11 & data2007$Day == 1 )
  # head(dailyTemps)
  #extract row with min column value
  # maxdailytemp<-dailyTemps[which.max(dailyTemps$LoadEst),]

#prepare o/p data frame for month of January Only
MaxTemp_hr.January<-data.frame(matrix(ncol = 12, nrow = 31))
k<-0
#Populate data frame with January max load values
for (j in 1:31){
  k=k+1
  dailyTemps1<-subset(data2007, subset = data2007$Month == 1 & data2007$Day == j )
  MaxTemp_hr.January[k,]<-dailyTemps1[which.max(dailyTemps1$LoadEst),]
}
head(MaxTemp_hr.January)
```

```
##     X1 X2 X3 X4 X5 X6 X7 X8 X9 X10   X11     X12
## 1 2007  1  1  2 20 64 66 73 54  NA 13514 1343790
## 2 2007  1  2  3  9 55 59 70 41  NA 13515 1478543
## 3 2007  1  3  4 20 68 71 76 58  NA 13516 1383801
## 4 2007  1  4  5 19 70 70 76 65  NA 13517 1401799
## 5 2007  1  5  6 20 71 73 77 61  NA 13518 1426328
## 6 2007  1  6  7 19 71 72 76 64  NA 13519 1409903
```

```
Jan.daily.maxLoad<-MaxTemp_hr.January$X12
write.table(Jan.daily.maxLoad, file = "January_daily.csv", sep = ",", col.names = NA)
Jan.daily.maxTime<-MaxTemp_hr.January$X5
write.table(Jan.daily.maxTime, file = "January_timely.csv", sep = ",", col.names = NA)

#Populate data frame with February max load values
MaxTemp_hr.Feb<-data.frame(matrix(ncol = 12, nrow = 28))
k<-0
for (j in 1:28){
  k=k+1
  dailyTemps1<-subset(data2007, subset = data2007$Month == 2 & data2007$Day == j )
  MaxTemp_hr.Feb[k,]<-dailyTemps1[which.max(dailyTemps1$LoadEst),]
}

Feb.daily.maxLoad<-MaxTemp_hr.Feb$X12
write.table(Feb.daily.maxLoad, file = "Feb_daily.csv", sep = ",", col.names = NA)
Feb.daily.maxTime<-MaxTemp_hr.Feb$X5
write.table(Feb.daily.maxTime, file = "Feb_timely.csv", sep = ",", col.names = NA)

#Populate data frame with March max load values
MaxTemp_hr.March<-data.frame(matrix(ncol = 12, nrow = 31))
k<-0
#Populate data frame with March max load values
for (j in 1:31){
  k=k+1
  dailyTemps1<-subset(data2007, subset = data2007$Month == 3 & data2007$Day == j )
  MaxTemp_hr.March[k,]<-dailyTemps1[which.max(dailyTemps1$LoadEst),]
}
head(MaxTemp_hr.March)
```

```
##     X1 X2 X3 X4 X5 X6 X7 X8 X9 X10   X11     X12
## 1 2007  3  1  5 20 74 74 78 68  NA 13573 1552792
## 2 2007  3  2  6 19 66 62 82 50  NA 13574 1309731
## 3 2007  3  3  7 21 57 55 70 52  NA 13575 1337662
## 4 2007  3  4  1  9 56 57 63 48  NA 13576 1437178
## 5 2007  3  5  2  7 40 43 51 27  NA 13577 1981450
## 6 2007  3  6  3  7 38 39 54 28  NA 13578 2084658
```

```
March.daily.maxLoad<-MaxTemp_hr.March$X12
write.table(March.daily.maxLoad, file = "March_daily.csv", sep = ",", col.names = NA)
March.daily.maxTime<-MaxTemp_hr.March$X5
write.table(March.daily.maxTime, file = "March_timely.csv", sep = ",", col.names = NA)

#Populate data frame with April max load values
MaxTemp_hr.April<-data.frame(matrix(ncol = 12, nrow = 30))
k<-0
#Populate data frame with April max load values
for (j in 1:30){
  k=k+1
  dailyTemps1<-subset(data2007, subset = data2007$Month == 4 & data2007$Day == j )
  MaxTemp_hr.April[k,]<-dailyTemps1[which.max(dailyTemps1$LoadEst),]
}
head(MaxTemp_hr.April)
```

```
##     X1 X2 X3 X4 X5 X6 X7 X8 X9 X10   X11     X12
## 1 2007  4  1  1 18 79 79 86 73  NA 13604 1716873
## 2 2007  4  2  2 17 81 82 88 76  NA 13605 1805283
## 3 2007  4  3  3 18 81 82 88 75  NA 13606 1879779
## 4 2007  4  4  4 18 83 84 88 78  NA 13607 1988547
## 5 2007  4  5  5 21 66 64 79 59  NA 13608 1446144
## 6 2007  4  6  6  9 54 55 65 41  NA 13609 1454248
```

```
April.daily.maxLoad<-MaxTemp_hr.April$X12
write.table(April.daily.maxLoad, file = "April_daily.csv", sep = ",", col.names = NA)
April.daily.maxTime<-MaxTemp_hr.April$X5
write.table(April.daily.maxTime, file = "April_timely.csv", sep = ",", col.names = NA)


#Populate data frame with November max load values
MaxTemp_hr.November<-data.frame(matrix(ncol = 12, nrow = 30))
k<-0
#Populate data frame with November max load values
for (j in 1:30){
  k=k+1
  dailyTemps1<-subset(data2007, subset = data2007$Month == 11 & data2007$Day == j )
  MaxTemp_hr.November[k,]<-dailyTemps1[which.max(dailyTemps1$LoadEst),]
}
head(MaxTemp_hr.November)
```

```
##     X1 X2 X3 X4 X5 X6 X7 X8 X9 X10   X11     X12
## 1 2007 11  1  5 18 78 79 83 73  NA 13818 1761639
## 2 2007 11  2  6 18 73 75 83 67  NA 13819 1502255
## 3 2007 11  3  7 18 72 72 77 69  NA 13820 1448736
## 4 2007 11  4  1 21 57 57 68 47  NA 13821 1362283
## 5 2007 11  5  2 21 57 57 68 47  NA 13822 1369082
## 6 2007 11  6  3 18 69 70 73 62  NA 13823 1404813
```

```
November.daily.maxLoad<-MaxTemp_hr.November$X12
write.table(November.daily.maxLoad, file = "November_daily.csv", sep = ",", col.names = NA)
November.daily.maxTime<-MaxTemp_hr.November$X5
write.table(November.daily.maxTime, file = "November_timely.csv", sep = ",", col.names = NA)

#Populate data frame with December max load values
MaxTemp_hr.December<-data.frame(matrix(ncol = 12, nrow = 31))
k<-0
#Populate data frame with December max load values
for (j in 1:31){
  k=k+1
  dailyTemps1<-subset(data2007, subset = data2007$Month == 12 & data2007$Day == j )
  MaxTemp_hr.December[k,]<-dailyTemps1[which.max(dailyTemps1$LoadEst),]
}
head(MaxTemp_hr.December)
```

```
##     X1 X2 X3 X4 X5 X6 X7 X8 X9 X10   X11     X12
## 1 2007 12  1  7 19 67 68 80 57  NA 13848 1469112
## 2 2007 12  2  1 20 68 69 77 61  NA 13849 1522290
## 3 2007 12  3  2 19 67 70 77 55  NA 13850 1574704
```

```
## 4 2007 12  4  3  8 45 45 69 34  NA 13851 1919963
## 5 2007 12  5  4  7 41 41 58 30  NA 13852 1999070
## 6 2007 12  6  5  8 50 49 61 35  NA 13853 1675242
```

```
December.daily.maxLoad<-MaxTemp_hr.December$X12
write.table(December.daily.maxLoad, file = "December_daily.csv", sep = ",", col.names = NA)
December.daily.maxTime<-MaxTemp_hr.December$X5
write.table(December.daily.maxTime, file = "December_timely.csv", sep = ",", col.names = NA)
```