Regreesion model through principal component analysis

Supria Paul

May 11, 2018

calling all library for running this code.

```
library(lubridate)
library(tidyverse)
library(ggpubr)
library(gridExtra)
library(tinytex)
library(FactoMineR)
```

R Markdown

creating function for reading several csv files at once time.

you have to use lubridate library for changing factor to date format.

I used ymd function which helps to change the factor format to pdf file.

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
batch_read <- function( pattern, recursive = FALSE, read_fun, ...) {
  data.files <- list.files( pattern = pattern, recursive = recursive)
  data <- lapply(paste0( data.files), read_fun, ...)
  data <- do.call("rbind", data)
  data
}
LAI.data<- batch_read (   pattern = "\\.csv" ,read_fun = read.csv, header = TRUE)

LAI.data$date<-ymd(LAI.data$date)
str(LAI.data)
```

Data processing for further analysis

```
LAI.Data.A = LAI.data[order(LAI.data$date),]
LAI.Data.A <- cbind(LAI.Data.A, Month=month(LAI.Data.A$date))
LAI.Data.A <- cbind(LAI.Data.A, Year=year(LAI.Data.A$date))
LAI.Data.A <- cbind(LAI.Data.A, Day=day(LAI.Data.A$date))

LAI.Data.A<- LAI.Data.A[-(1:2)]
LAI.Data.A<- LAI.Data.A[-3]
```

Data processing for further analysis

```
Weather.data<-read.csv('weather_data/weatherdata_417_716.csv',header = TRUE)
Weather.data$Date<-mdy(Weather.data$Date)
Weather.data.chg<- cbind(Weather.data, date=Weather.data$Date)
str(Weather.data)

## combining all weather parameters and LAI data
Final_datasets <- merge(Weather.data.chg,LAI.Data.A,by=c("date"))
Final_datasets<-Final_datasets[-1]
```

Data processing for further analysis

```
###creating subset where will be only may month data and no missing value and also there will be no LAI
MAy.compare<-Final_datasets%>%filter(Month==5,!is.na(LAI),!LAI==0)
MAy.compare <- MAy.compare[-1]
MAy.compare <- MAy.compare[-(8:10)]
may.compare.mean <- mean(MAy.compare,na.rm =TRUE)

### creating standarize value for each parameters.
Mean.LAI <-MAy.compare %>% summarize(mean_LAI = mean(LAI, na.rm = TRUE),
                                     sd_LAI= sd(LAI, na.rm = TRUE))
Mean.mxt <-MAy.compare %>% summarize(mean_MxT = mean(Max.Temperature, na.rm = TRUE),
                                     sd_MxT = sd(Max.Temperature, na.rm = TRUE))
Mean.mT <-MAy.compare %>% summarize(mean_MT = mean(Min.Temperature, na.rm = TRUE),
                                    sd_MT = sd(Min.Temperature, na.rm = TRUE))
Mean.Solar <-MAy.compare %>% summarize(mean_S = mean(Solar, na.rm = TRUE),
                                       mean_Sd = sd(Solar, na.rm = TRUE))
Mean.HR <-MAy.compare %>% summarize(mean_HR = mean(Relative.Humidity, na.rm = TRUE),
                                    mean_sd = sd(Relative.Humidity, na.rm = TRUE))
Mean.wd <-MAy.compare %>% summarize(mean_wd = mean(Wind, na.rm = TRUE),
                                    mean_sd = sd(Wind, na.rm = TRUE))
Mean.pp <-MAy.compare %>% summarize(mean_pp = mean(Precipitation, na.rm = TRUE),
                                    mean_pp = sd(Precipitation, na.rm = TRUE))


### standarize parameters will be (mean value of each varibles - values of each varibles) / standard de
MAy.compare$Max.Temperature<-
  (-MAy.compare$Max.Temperature+Mean.mxt$mean_MxT)/Mean.mxt$sd_MxT
MAy.compare$Min.Temperature<-
  (-MAy.compare$Min.Temperature+Mean.mT$mean_MT)/Mean.mT$sd_MT
MAy.compare$Precipitation<-
  (-MAy.compare$Precipitation+Mean.pp$mean_pp)/Mean.pp$mean_pp
MAy.compare$Wind<-
  (-MAy.compare$Wind+Mean.wd$mean_wd)/Mean.wd$mean_sd
MAy.compare$Relative.Humidity<-
  (-MAy.compare$Relative.Humidity+Mean.HR$mean_HR)/Mean.HR$mean_sd
MAy.compare$Solar<-
  (-MAy.compare$Solar+Mean.Solar$mean_S)/Mean.Solar$mean_Sd
MAy.compare$LAI<-
  (-MAy.compare$LAI+Mean.LAI$mean_LAI)/Mean.LAI$sd_LAI
```

Data processing for further analysis

```
#Prepare train and test data for regression models
numeric <- colnames(MAy.compare)
numeric <- numeric[-7]
Target<-c("LAI")

set.seed(42)
train <-sample(nrow(MAy.compare), 0.7*nrow(MAy.compare))
test<-setdiff(seq_len(nrow(MAy.compare)), train)
```

```
Data_for_PCA<-MAy.compare[,numeric]
### finding co-relation of each weather variables
pca1<-PCA(Data_for_PCA,ncp = 2)
plot(pca1)
PCA_data<-as.data.frame(cbind(MAy.compare[train,c(Target,numeric)],pca1$ind$coord[train,]))
Step_PCA_Reg<-step(lm(LAI~.,data = PCA_data))
summary(Step_PCA_Reg)
PCA_Estimate <- predict(Step_PCA_Reg, type="response", newdata=cbind(MAy.compare[test,c(Target,numeric)]
Observed <- subset(MAy.compare[test,c(numeric,Target)],select = Target)
### calculate R-squared value
format(cor(PCA_Estimate, Observed$LAI)^2, digits=4)
```

Data processing for further analysis

```
## Regression analysis based whole data sets
```

```
Final_datasets_total <-Final_datasets%>%filter(!is.na(LAI),!LAI==0)
Final_datasets_total <- Final_datasets_total[-1]
Final_datasets_total <- Final_datasets_total[-(8:10)]
```

```
#Prepare train and test data for regression models
numeric <- colnames(Final_datasets_total)
numeric <- numeric[-7]
Target<-c("LAI")
set.seed(42)
train <-sample(nrow(Final_datasets_total), 0.7*nrow(Final_datasets_total))
test<-setdiff(seq_len(nrow(Final_datasets_total)), train)
```

```
Data_for_PCA<-Final_datasets_total[,numeric]
pca1<-PCA(Data_for_PCA,ncp = 2)
plot(pca1)
PCA_data<-as.data.frame(cbind(Final_datasets_total[train,c(Target,numeric)],pca1$ind$coord[train,]))
```

```
Step_PCA_Reg<-step(lm(LAI~.,data = PCA_data))
summary(Step_PCA_Reg)
PCA_Estimate <- predict(Step_PCA_Reg, type="response", newdata=cbind(Final_datasets_total[test,c(Target
Observed <- subset(Final_datasets_total[test,c(numeric,Target)],select = Target)
format(cor(PCA_Estimate, Observed$LAI)^2, digits=4)
```