SCHOOL OF MATH DEPARTMENT

Statistical Analysis on Animal Sleeping Data

Group memeber: Liu Ping, Guo Feifei

April 10, 2017

1 ABSTRACT

We did some statistical analysis on the animal sleeping dataset. We firstly use many different methods to fill the missing value. Then we did clustering analysis of the mammals. Based on the classification, we find more interesting things, such as the body size has the most important influence on sleep time for the mammals who want to sleep long. That may contradict the common sense that the danger is of most importance.

2 INTRODUCTION

The animal sleeping dataset is in the form of 60×10 matrix, containing 60 different species of mammals and 10 features of these mammals(slowWaveSleep,dreamSleep,sleep,body,brain,life, gestation,predation,exposure,danger). Each column contains the value of a feature of all the mammals.

Based on our statistical analysis we want to figure the relation between slowWaveSleep,dream Sleep ,sleep and body size, danger or other features. Besides this theme, we want find some interesting information also.

This report presents some statistical analysis of the dataset which focus on correlation and clustering analysis of the features and mammals. We choose some statistic methods to pre-process the dataset to fill the missing value. And use many different ways to fill the missing value, such as KNN method and lasso. Then we make a classification of the animals based on their body size and danger. We then have four categories and each of them represents a special feature. Then the four categories have great difference in sleep time. Besides this, we find some other interesting information.

This report is organized as follows. In section 3 we use many different methods to fill the missing value. And in section 4 we make a classification of the mammals based on their body size and danger. Then we get some interesting information from the classification. Finally we give a summary.

3 Data processing

3.1 Introduction to dataset

Variable	Description
BodyWt	body weight (kg)
BrainWt	brain weight (g)
NonDreaming	slow wave ("nondreaming") sleep (hrs/day)
Dreaming	paradoxical ("dreaming") sleep (hrs/day)
TotalSleep	total sleep, sum of slow wave and paradoxical sleep (hrs/day)
LifeSpan	maximum life span (years)
Gestation	gestation time (days)
Predation	predation index (1-5) 1 = minimum (least likely to be preyed upon); 5 = maximum (most likely to be preyed upon)
Exposure	sleep exposure index (1-5) 1 = least exposed (e.g. animal sleeps in a well-protected den); 5 = most exposed (e.g. animal sleeps in a badly-protected den)
Danger	overall danger index (1-5) (based on the above two indices and other information) 1 = least danger (from other animals); 5 = most danger (from other animals)

Table 3.1: Feature description

3.2 Pre-Processing

3.2.1 Handle missing value

Check the situation of the missing value, and their distribution. If a variable data loss rate is too high, this variable may not have great research significance. It may like the followed figure.

species	slow wave sleep	dreamsleep	body	brain	life
0	14	12	4	0	0
gestation	body	predation	exposure	danger	
4	4	0	0	0	

3.2.2 OTHER PROCESSING

We observe the data and take the log (x + 0.1) on variables which have large scope in value range, such as brain and body, gestation. The distribution of each variable are in the following table,

feature	brain	life	gestation	body	predation	exposure	danger
0%	0.1	2.0	12.0	0.0	1	1	1
25%	4.2	6.6	35.8	0.6	2	1	1
50%	17.2	15.1	79.0	3.3	3	2	2
75%	166.0	27.8	207.5	48.2	4	4	4
100%	5712.0	100.0	645.0	6654.0	5	5	5

Table 3.2: Distribution of each variable

3.3 METHOD TO HANDLE MISSING VALUE

3.3.1 DIRECT DELETE

Sometimes,directly deleting samples containing missing value is the most effective way. But the premise is that less proportion of missing data contained in the dataset and the missing data is random. Base on the premise, influence on the result of the analysis is little after removing the missing value.

3.3.2 USE THE MEAN/MEDIAN/MODAL FILL

The advantage of this method is that it will not reduce the sample information and processing is simple;The disadvantage is the result is skewed when missing data is not random.

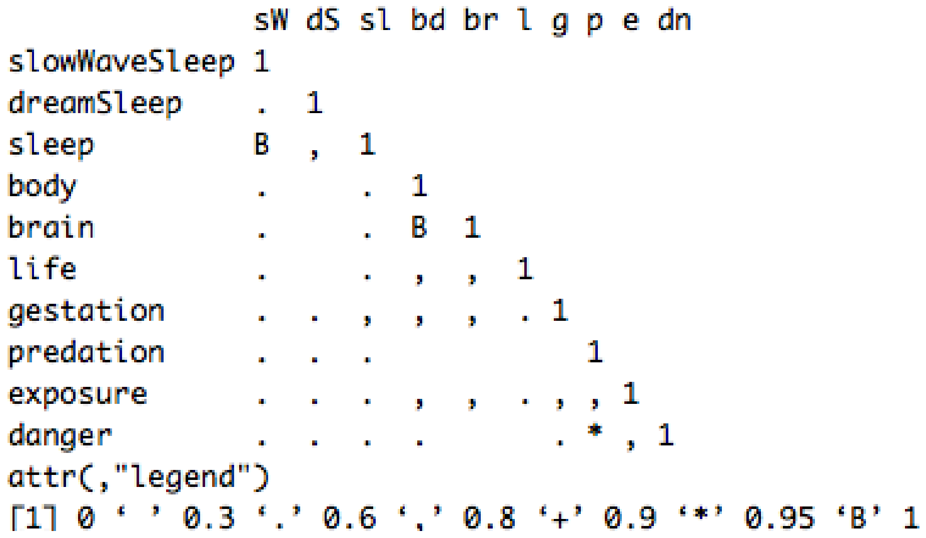
3.3.3 INVESTIGATE THE SIMILARITY OF SAMPLES

Utilizing the similarity between multiple variables to fill the missing value. There are varies indexes to determine the similarity and the common metric is Euclidean distance. Common method is KNN method. For the missing data, we find k samples which are closest with the missing data according to the Euclidean distance. And then we use the weighted average of these samples to fill the missing data. The advantage of this method is that we can obtain all of the missing data using this method one time. The disadvantage is that this method will not have an effective result for the factor variables.

3.3.4 INVESTIGATE THE SIMILARITY OF VARIABLES

Investigate the relationship between variables and find the two variables that have a high correlation. Then find the linear regression relation between them and compute and fill the missing value by the linear regression relation in the end.

According to the relation between the seven variables, we fill the missing value of variable life and gestation by the KNN method.



```
              sW dS sl bd br l g p e dn
slowWaveSleep 1
dreamSleep    .  1
sleep         B  ,  1
body          .     .  1
brain         .     .  B  1
life          .     .  ,  ,  1
gestation     .  .  ,  ,  ,  . 1
predation     .  .  .            1
exposure      .  .  .  ,  ,  . , , 1
danger        .  .  .  .       . * , 1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```

Figure 3.1: correlation of all variables

3.3.5 FILL THE VALUE OF SLEEP

According to the correlation of these variables, we predict the missing data of sleep using different regression methods. We totally use five methods consist of simple linear regression method, lasso method, decision tree regression, bagging regression, random forest regression. Firstly, we predict missing data using linear regression. The related information is followed.

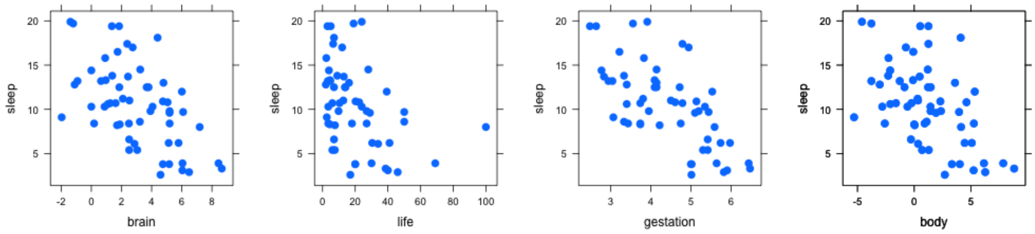


Figure 3.2: variable have high relation with sleep

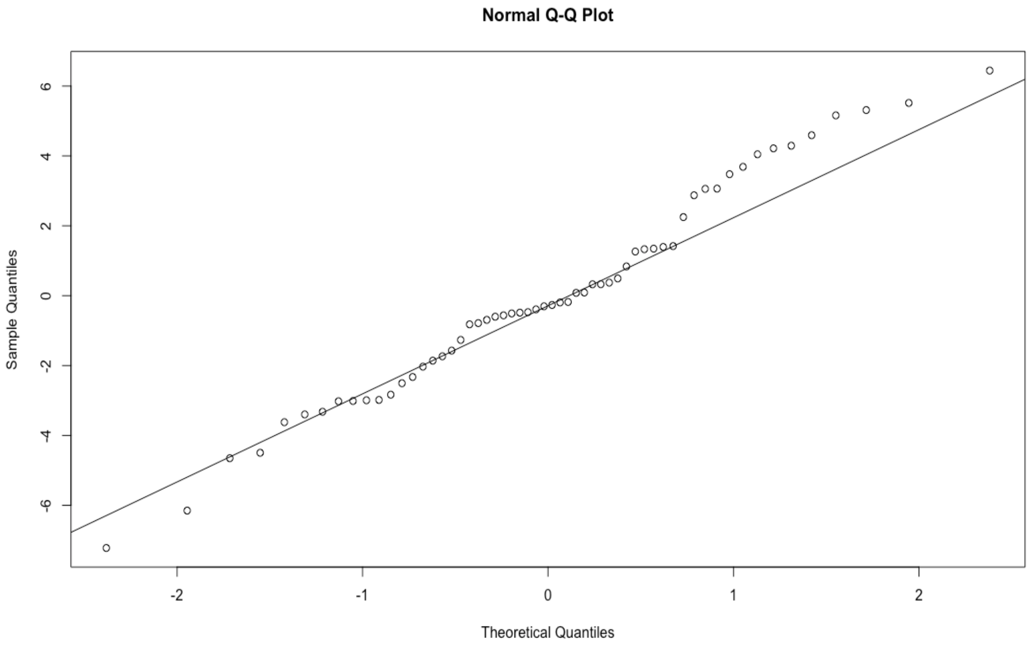
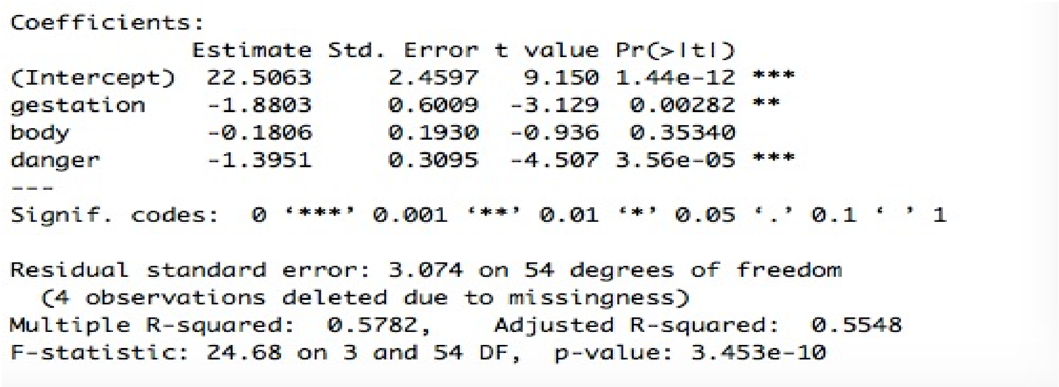


Figure 3.3: residuals plot of the linear regression



```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  22.5063     2.4597   9.150 1.44e-12 ***
gestation    -1.8803     0.6009  -3.129  0.00282 ** 
body         -0.1806     0.1930  -0.936  0.35340    
danger       -1.3951     0.3095  -4.507 3.56e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.074 on 54 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared:  0.5782,	Adjusted R-squared:  0.5548 
F-statistic: 24.68 on 3 and 54 DF,  p-value: 3.453e-10
```

Figure 3.4: linear regression coefficient

mamamls	21	31	41	62
sleep	3.132220	9.709187	3.088408	14.018722

Table 3.3: predNA of sleep using lm method

It is a very straightforward approach for predicting a quantitative response Y on the basis of a single predictor variable X. However, the effect of collinearity of those variables is to make the regression coefficients unreliable. Therefore, we consider using the lasso method to select variables and do regression. Related results are followed.

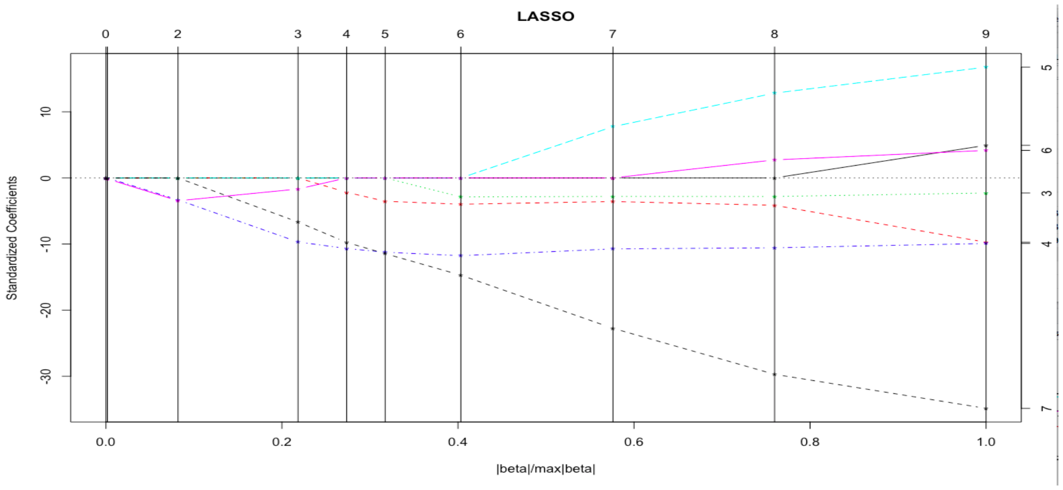


Figure 3.5: the coefficients in the lasso regression

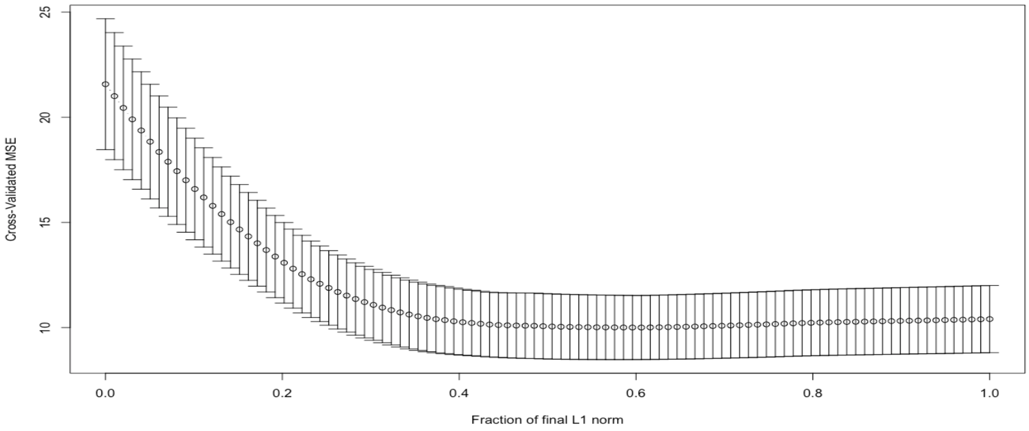


Figure 3.6: the value of CV in lasso regression

mamamls	21	31	41	62
sleep	3.508987	8.105562	3.548290	16.523382

Table 3.4: predNA of sleep using lasso method

And then we use decision tree method to fill the missing data. Firstly, we begin with a simple tree, but this tree maybe more complex and need to be prune it in order to obtain a subtree. Intuitively, our goal is to select a subtree that leads to the lowest test error rate using K-fold cross validation method.

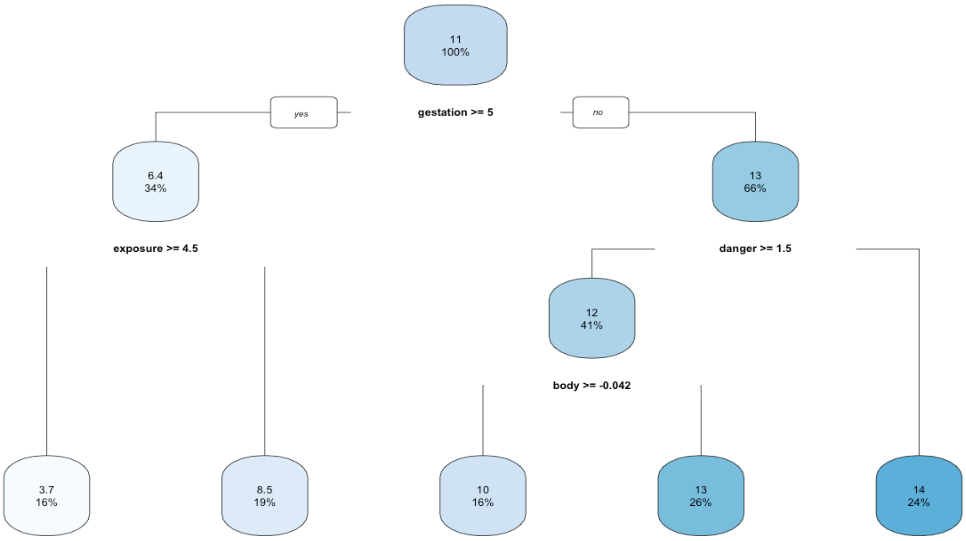


Figure 3.7: regression tree analysis for the animal sleep data

mamamls	21	31	41	62
sleep	3.508987	8.105562	3.548290	16.523382

Table 3.5: predNA of sleep using decision tree

And then we consider use bagging, random forests use trees as building blocks to construct more powerful prediction models. Bagging method is the average of many trees obtained by bootstrap method. Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees. As in bagging, we build a number of decision trees on bootstrapped training samples. But when building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. All the filled results are shown in the followed table.

mamamls	21	31	41	62
sleep	4.918694	9.282358	4.918694	12.916063

Table 3.6: predNA of sleep using bagging

mamamls	21	31	41	62
sleep	3.574998	9.519205	3.923274	14.123291

Table 3.7: predNA of sleep using SVR

mamamls	21	31	41	62
sleep	3.826617	7.999091	4.003553	12.032031

Table 3.8: predNA of sleep using random forest

4 CLUSTERING ANALYSIS OF ANIMALS

4.1 SUPPORT VECTOR MACHINE(SVM)

Support vector machine is the method solving the following optimization problem:

$$\min_{\omega,b,\xi} \frac{1}{2}\langle\omega,\omega\rangle + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } y_i(\langle\omega,\varphi(x_i)\rangle + b) \geq 1-\xi_i, \forall i = 1,2,\cdots,n$$

classify function is:

$$F(x) = sgn(\langle\omega,\varphi(x_i)\rangle + b)$$

This optimization problem satisfies KKT condition, so the optimization solution of it's dual problem is the optimal solution to the original problem, and:

$$\omega^* = \sum_{i=1}^{n}\omega_i\varphi(x_i)$$

If there exists

$$K(x,y) = \langle\varphi(x),\varphi(y)\rangle,$$

Then the dual problem is:

$$\max_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$

$$\text{subject to } \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n}\alpha_i y_i = 0 \end{cases}$$

classify function is:

$$F(x) = \sum_{i=1}^{n}\alpha_i^* y_i K(x,x_i) + b^*$$

Common kernel functions:

Linear Kernel: $K(x,y) = x^T y$

Radial Basis Kernel: $K(x,y) = \exp(-\gamma||x-y||^2)$

Polynomial Kernel: $K(x,y) = (\gamma x^T y + \text{constant})^d$

Sigmoid Kernel: $K(x,y) = \tanh(\gamma x^T y + \text{constant})$

4.2 Clustering analysis

Although we have find some general correlation between some different variables, such as the correction between slow wave sleep and danger, we want to gain more specific information on the relation between sleep time and body and danger. We firstly determine 4 category as the followed table shown.

category	body	danger
I	large	high
II	large	low
III	small	high
IV	small	low

We firstly choose four special species as a training set and then cluster all the mammals with completed data using the above support vector machine method. Then we get our clustering result.

category	mammals
I	Asian elephant Baboon Brazilian tapir Cow Goat Horse Patas monkey Pig Rhesus monkey
II	Chimpanzee Gray seal Human Sheep
III	African giant pouched rat Chinchilla Ground squirrel Guinea pig Lesser short-tailed shrew Mouse Musk shrew Rabbit Rat Rock hyrax (Procavia hab) Tree hyrax Vervet
IV	Big brown bat Cat Eastern American mole Echidna European hedgehog Galago Golden hamster Little brown bat N. American opossum Nine-banded armadillo Owl monkey Phanlanger Red fox Rock hyrax (Hetero. b) Tenrec Tree shrew Water opossum

Then we compare the average sleeping time among the four categories and have the result shown in the followed table.

category	average of sleep time
I	6.6
II	6.925
III	10.05
IV	14.076

From the above table we can obviously see that the average sleep time increase as the category numbers increase. And obviously if the mammal has large body and high danger then it's sleeping time is very short. And if the mammal has a small body and low danger then it's sleeping time is very long. What is more interesting is the difference between category II and III. It indicates that the body size may be the most important feature in deciding the sleep time. Besides this observation, we did more specific analysis based on this classification in the followed section.

4.3 Analysis of specific category

Observing the data in each category, we get an interesting idea that the category II and IV both have low danger but the mammals in category IV obviously have very long sleep time. This indicate that body size is an important feature which have great influence on the sleep time especially for the animals have low danger. We analyze the mammals who have more than 10 hours sleeping time and more than 15 hours sleeping time. As the followed figure shown.

species	sleep	body	danger
Big brown bat	19.7	0.023	1
Chinchilla	12.5	0.425	4
European hedgehog	10.7	0.785	2
Galago	10.7	0.2	2
Golden hamster	14.4	0.12	2
Ground squirrel	13.8	0.101	3
Little brown bat	19.9	0.01	1
Mouse	13.2	0.023	3
Musk shrew	12.8	0.048	3
N. American opossum	19.4	1.7	1
Phanlanger	13.7	1.62	2
Rat	13.2	0.28	3
Tenrec	13.3	0.9	2
Tree shrew	15.8	0.104	2
Nine-banded armadillo	17.4	3.5	1
Owl monkey	17	0.48	2
Patas monkey	10.9	10	4
Vervet	10.3	4.19	4
Cat	14.5	3.3	1
Water opossum	19.4	3.5	1

Table 4.1: mammals sleep time ≥ 10

From above figure we can directly see that the mammals who have more that 10 hours sleep usually have tiny body but they may not have low danger. This is very interesting. It shows that the feature that determines if the mammals have a long sleep time is it's body size.

And we should explain here why we do not use the correlation analysis here. Cause the danger data is an integer created by human but the body size is real value which is not integer. So the correlation analysis may not show the real relation between sleep time and body and danger. Using the table directly is more accurate.

The followed table is about the mammals who sleep more than 15 hours.

species	sleep	body	danger
Big brown bat	19.7	0.023	1
Little brown bat	19.9	0.01	1
N. American opossum	19.4	1.7	1
Nine-banded armadillo	17.4	3.5	1
Owl monkey	17	0.48	2
Tree shrew	15.8	0.104	2
Water opossum	19.4	3.5	1

Table 4.2: mammals sleep time ≥ 15

From the result we can see that if the mammals have very long sleep time(more than 15 hours), then it must have small body and low danger. That is still very interesting. It means that, different from the mammals sleep around 10 hours per day, the mammals who have a very long sleep must be in less danger. Certainly they should be very tiny.

5 Summary

We use some methods in statistic and machine learning to study some useful information hidden in the dataset. We fill the missing value using many different methods. And then we did clustering analysis on the mammals and gain interesting information based on this idea.

Appendices

```
#data processing
\textbf{setwd("/Users ")
sleep <- read.csv('sleep.csv');

names(sleep)[names(sleep) == 'sleepExposure'] = 'exposure'

# Count NAs
nNAs <- sapply(is.na(sleep), sum)
nNAs <- apply(is.na(sleep), 2, sum)

newX <- c('brain', 'life', 'gestation', 'predation','exposure')
allX <- c(newX[1:3], 'body', newX[4:5], 'danger')
round(sapply(sleep[, newX], quantile, na.rm = TRUE), 1)

sleepR2 <- rep(NA, 7)
names(sleepR2) <- allX
for (i in allX[1:4]) {
z <- lm(sleep$sleep ~ log(sleep[[i]]))
sleepR2[i] <- summary(z)$r.squared
}
for (i in allX[5:7]) {
z <- lm(sleep$sleep ~ sleep[[i]])
sleepR2[i] <- summary(z)$r.squared
}

sleepLog <- sleep
for (i in c('brain', 'gestation', 'body')) {
sleepLog[[i]] <- log(sleep[[i]])
}

symnum(cor(sleepLog[,4:11],use="complete.obs"))
vacor=cor(sleepLog[,4:11],use="complete.obs")
write.csv(vacor,file="variablecorr.csv")

plotList <- list()
for (i in allX) {
plotList[[i]] <- xyplot(sleepLog$sleep ~ sleepLog[[i]],
xlab = i, ylab = 'sleep',
pch = 16, cex = 1.25)
}
png('onePredictor.png', height = 250, width = 1000)
```

```
for (i in 1:4) {
print(plotList[[i]], position = c(.25 * (i-1), 0, .25*i, 1), more = TRUE)
}
print(plotList[[i]], position = c(.75, 0, 1, 1))
graphics.off()

png('twoPredictor.png', height = 250, width = 1000)
for (i in 1:4) {
print(plotList[[i]], position = c(.25 * (i-1), 0, .25*i, 1), more = TRUE)
}
print(plotList[[i]], position = c(.75, 0, 1, 1))
graphics.off()

png('twoPredictor.png', height = 650, width = 700)
print(plotList[[5]])
graphics.off()

png('two2Predictor.png', height = 650, width = 700)
print(plotList[[6]])
graphics.off()

png('two3Predictor.png', height = 650, width = 700)
print(plotList[[7]])
graphics.off()

library(Hmisc)
## mean method
library(survival)
library(Formula)
library(ggplot2)
treated$ptratio=impute(treated$ptratio, mean)  #æŔŠèąĕåİĞåĂij
treated$ptratio=impute(treated$ptratio, median) #æŔŠèąĕäÿŋä¡ĐæŢř
treated$ptratio=impute(treated$ptratio, 20.2)  #åąńåĚĚçĽźåőŽåĂij

## knn method
library(lattice)
library(grid)
library(DMwR)
dataknn=knnImputation(sleepLog[,5:11], k=10, meth="weighAvg") ;
sleepLog[,5:11]=dataknn;

## linear regression method
library(MASS)
z <- lm(sleep ~ gestation + body + danger, data = sleepLog)
```

```
summary(z)
# Get fitted values for species with NA for sleep$sleep
indx <- which(is.na(sleepLog$sleep))
predNA <- predict(z, newdata = sleepLog[indx, ])
predAll <- predict(z, newdata = sleepLog)

# Plot fitted values against each of the predictors
plotList <- list()
for (i in c('gestation', 'body', 'danger')) {
plotList[[i]] <- xyplot(predAll ~ sleepLog[[i]],
ylab = 'sleep',
xlab = i,
pch =  16, cex = 1.25,
panel = function(x, y, ...) {
panel.grid(h=-1, v=-1)
panel.xyplot(x, y, ...)
panel.points(sleepLog[[i]][indx], predNA,
col = 'red', pch = 16, cex = 1.25)
})
}
png('fittedNA.png', height = 250, width = 800)
print(plotList[[1]], position = c(0, 0, 1/3, 1), more = T)
print(plotList[[2]], position = c(1/3, 0, 2/3, 1), more=T)
print(plotList[[3]], position = c(2/3, 0, 1, 1))
graphics.off()

# Plot observed values against each of the predictors, but
# use the fitted values for the outcomes with NA.
plotList <- list()
for (i in c('gestation', 'body', 'danger')) {
plotList[[i]] <- xyplot(sleep$sleep ~ sleepLog[[i]],
ylab = 'sleep',
xlab = i,
pch =  16, cex = 1.25,
panel = function(x, y, ...) {
panel.grid(h=-1, v=-1)
panel.xyplot(x, y, ...)
panel.points(sleepLog[[i]][indx], predNA,
col = 'red', pch = 16, cex = 1.25)
})
}
png('observedNA.png', height = 250, width = 800)
print(plotList[[1]], position = c(0, 0, 1/3, 1), more = T)
print(plotList[[2]], position = c(1/3, 0, 2/3, 1), more=T)
print(plotList[[3]], position = c(2/3, 0, 1, 1))
graphics.off()


## lasso method
library(lars)
sleep1=sleepLog[,4:11];
sleep2<-na.omit(sleep1)#è£ŤåŻđåŐżæŐĽNAåĂijçŽĎåŘŚéĞŔ
x=as.matrix(sleep2[,2:8]);
y=as.matrix(sleep2[,1])
x2=x;
laa=lars(x2,y);
plot(laa)
summary(laa)
cva=cv.lars(x2,y,K=10);
best=cva$index[which.min(cva$cv)]
coef=coef.lars(laa,mode="fraction",s=best)
min(laa$Cp)

z <- lm(sleep ~ brain+life+gestation+predation+exposure+danger, data = sleepLog)
summary(z)
indx <- which(is.na(sleepLog$sleep))
predNA <- predict(z, newdata = sleepLog[indx, ])
# fillPO4<-function(x)
#   {  if(is.na(x)) return(NA) else return(-0.19448606*x[[1]]-0.02011583*x[[2]]-1.39828157
# algae[is.na(algae$PO4),"PO4"]<-sapply(sleepLog[indx,6:11],fillPO4)

## decision method
library(rpart)
library(rpart.plot)
sleepall=rpart(sleep~.,data=sleepLog[,4:11], na.action=na.omit, method="anova")
sleep_pred=predict(sleepall,sleepLog)
a=rpart(y~.,w)
rpart.plot(a,type=2,faclen=T)
rpart.plot(sleepall,type=2,faclen=T)
indx <- which(is.na(sleepLog$sleep))
predNA <- predict(sleepall, newdata = sleepLog[indx, ])
predAll <- predict(sleepall, newdata = sleepLog[-indx,])
rsquare=1-(sum((sleepLog[-indx,4]-predAll)^2))/(sum((sleepLog[-indx,4]-mean(sleepLog[-indx

##bagging method
library(ipred)
sleepall=bagging(sleep~.,data=sleepLog[,4:11])
predNA <- predict(sleepall, newdata = sleepLog[indx, ])
```

```
predAll <- predict(sleepall, newdata = sleepLog[-indx,])
rsquare=1-(sum((sleepLog[-indx,4]-predAll)^2))/(sum((sleepLog[-indx,4]-mean(sleepLog[-indx

## random forest method
library(randomForest)
sleepall=randomForest(sleep~.,data=sleepLog[,4:11],na.action=na.omit)
predNA <- predict(sleepall, newdata = sleepLog[indx, ])
predAll <- predict(sleepall, newdata = sleepLog[-indx,])
rsquare=1-(sum((sleepLog[-indx,4]-predAll)^2))/(sum((sleepLog[-indx,4]-mean(sleepLog[-indx


library(kernlab)
sleepall=ksvm(sleep~.,data=sleepLog[,4:11],na.action=na.omit)

library(mice)
sleepall=mice(treated[,!names(treated) %in% "medv"], method="rf")miceOutput<-complete(mice
sum(is.na(miceOutput))


sleepall=randomForest(sleep~.,data=sleepLog[,4:11],na.action=na.omit)
indx <- which(is.na(sleepLog$sleep))
#predNA <- predict(sleepall, newdata = sleepLog[indx, ])
sleepLog$sleep[is.na(sleepLog$sleep)] =predict(sleepall, newdata = sleepLog[indx, ])

sleepall=randomForest(slowWaveSleep~.,data=sleepLog[,c(2,5:11)],na.action=na.omit)
indx=which(is.na(sleepLog$slowWaveSleep))
sleepLog$slowWaveSleep[is.na(sleepLog$slowWaveSleep)] =predict(sleepall, newdata = sleepLo

indx=which(is.na(sleepLog$dreamSleep))
sleepLog$dreamSleep[is.na(sleepLog$dreamSleep)]=sleepLog[indx, 4]-sleepLog[indx, 2]

sleep[,2:4]=sleepLog[,2:4]
sleep[,7:8]=sleepLog[,7:8]
write.csv(sleep,file="sleepcomplete.csv")

## classification
sleep=read.csv("sleepcomplete1.csv")
data102=sleep[,c(1,2,3,5,6,7,8,9,10,11)];
library(factoextra)
library(ggplot2)
set.seed(1234)
cluster2=fviz_nbclust(data102[,-1],kmeans,method="wss")
geom_vline(xintercept = 5,linetype=2)
km.res=kmeans(data102[,-1],5)
fviz_cluster(km.res,data=data102[,-1])
km.res$cluster

datatest=data102$species[which(km.res$cluster==1)];
a1=names(datatest)
datatest2=data102$species[which(km.res$cluster==2)];
a2=names(datatest2)
datatest3=data102$species[which(km.res$cluster==3),];
a3=names(datatest3)
datatest4=data102$species[which(km.res$cluster==4),];
a4=names(datatest4)
datatest5=data102$species[which(km.res$cluster==5),];
a5=names(datatest5)
}

#svm
import numpy as np
import pandas as pd
import math
data = pd.read_csv("animal4.csv",header= None)
A=data
from sklearn import svm
X=[[0.023,1,1],[0.425,5,4],[62,1,1],[10.55,4,4]]
y=[0,1,2,3]
clf=svm.SVC()
clf.fit(X,y)
res=clf.predict(A)
np.set_printoptions(threshold='nan')
print(res)
```