**用模特卡洛模拟求e,sqrt(2),sqrt(3),ln(2)**

MC.ln2 <- function(n){

x <- runif(n,1,2)

y <- runif(n, 0,1)

return(sum(y< 1/x) /n)

}

MC.ln2(1000) # 一次模拟

#多次模拟以后求平均值

ys <- replicate(1000, MC.ln2(1000)) #对函数MC.ln2重复1000次，放回长度为1000的一个向量

mean(ys)

#----我的方法

N <- 10000

T <- runif(N)

mean(1/(1+T)) #一次模拟 ln2

mean(1/sqrt(2\*T)) #一次模拟 sqrt(2)

mean(3/2/sqrt(3\*T)) #一次模拟 sqrt(3)

sum(1/ sapply(0:N, factorial)) #一次计算e

mean(N/factorial(N\*T)) #一次计算e

支持向量机分类预测

rm(list=ls() )

require(e1071) #载入函数包

#---载入数据

Data <- read.csv('result.csv')

Data[,'C'] <- as.factor(Data[,'C'])

#----设置训练集和测试机

set.seed(123)

temp1 <- sample(1:dim(Data)[1], 100)

TrainData <- Data[temp1,] #随机抽取100个观测值作为训练集

TestData <- Data[-temp1,] #余下的作为测试集

#----开始训练

model <- svm(C ~ ., data = TrainData )#SVM函数有公式形式和数据形式，此为公式形式。

print(model)#输出显示SVM模型结果

summary(model)#汇总

# 用训练样本进行检验（test with train data）

pred <- fitted(model)

# 检验精确度（Check accuracy）:

(ConfuseMatrix<-table(pred, TrainData$C)) #构造混淆矩阵

prop.table(ConfuseMatrix,2) #按列计算比例，对角线上值为各类的预测精度

# 用测试集进行预测

x <- TestData[,1:8]

TePred <- predict(model, x)

# 检验精确度（Check accuracy）:

(ConfuseMatrix<-table(TePred , TestData$C)) #构造混淆矩阵

prop.table(ConfuseMatrix,2) #按列计算比例，对角线上值为各类的预测精度

特征变量提取并建立随机森林模型预测

library(ggplot2) # load the ggplot2 package

library(lubridate) # load the lubridate package:use to handle date

library(randomForest) # load the randomForest package

set.seed(1) # set a random seed in case of the random numbers' uniformity

train <- read.csv('../input/train.csv') # read the data from the csv file

test <- read.csv('../input/test.csv')

extractFeatures <- function(data) { # define a function whose arguement is data

## set feature variables

features <- c('season','holiday','workday','weather','temp','atemp','humidity','windspeed','hour')

data$hour <- hour(ymd\_hms(data$datetime)) # add another variable hour to the data

return(data[,features]) # return the features variables

}

trainFea <- extractFeatures(train) # extract the features variables from train data

testFea <- extractFeatures(test) # extract the features variables from test

## create a dataframe named submission which contains two elememts,datetime and count.

submission <- data.frame(datetime=test$datetime,count=NA)

for(i\_year in unique(year(ymd\_hms(test$datetime)))){

for(i\_month in unique(month(ymd\_hms(test$datetime)))){

cat('Year:', i\_year, '\tMonth:',i\_month,'\n')

testLocs <-year(ymd\_hms(test$datetime)) == i\_year & month(ymd\_hms(test$datetime)) == i\_month

testSubset <- test[testLocs,]

trainLocs <- ymd\_hms(train$datetime) <= min(ymd\_hms(testSubset$datetime))

rf <- randomForest(extractFeatures(train[trainLocs,]), train[trainLocs,'count'], ntree= 50)

submission[testLocs, 'count'] <- predict(rf, extractFeatures(testSubset))

}

}

联列表显著性检验

M <- as.table(rbind(c(762, 327, 468), c(484, 239, 477))) #将矩阵转化为联列表

dimnames(M) <- list(gender = c("M","F"),

party = c("Democrat","Independent", "Republican")) #按联列表格式命名

(Xsq <- chisq.test(M)) # Prints test summary #皮尔森卡方检验，是否显著

Xsq$observed # observed counts (same as M)

Xsq$expected # expected counts under the null

Xsq$residuals # Pearson residuals

Xsq$stdres # standardized residuals

Data1 <- read.csv('研究生教育满意度调查数据.csv')

Matr <- Data1[,2:4] # 提取变量

M1 <- as.matrix(Matr) #转化为矩阵

M2 <- as.table(M1) #转化为连列表

M2 #输出该连列表

Xsq2 <- chisq.test(M2)

"Mon Jun 08 14:14:26 2015"

记忆流年鑫人（河北联合大学）

问题提出  
一部m页的著作，总共有n个印刷错误，试利用poison过程近似求出连续x页无错误的概率。  
问题分析  
poison过程要想得出概率，需要知道印错的强度휆，本题휆=n/m 。假设从第k页出现错误，则푃[푁(푎+푥)−푁(푎)=0]=푒^(−휆∗푥)，  
求大神帮我写一个R语音程序，实现我改变x的值，p的值相应改变  
万谢

上边乱码了，不好意思哈。 poison过程要想得出概率，需要知道印错的强度휆，本题휆=n/m 。假设从第푎页出现错误，则푃=푒^(−휆∗푥) 就是写一个R程序，可以自己设定m和n的值，算出휆，给出任意的x，算出概率P。  
大神，求助啊，对于你来说应该不难的

求解代码：

#题目是利用possion分布对二项分布做近似

#-------帮助贴吧

m <- 5000 #设m为5000页

n <- 5 # n为5

x <- 5 # 假设x=5，这里x可以取成一个向量，同时求多个分布概率。

ppois(x, lambda=n)

##比如下面求多个x的概率

x <- 1:100

ppois(x, lambda=n)

"Mon Jun 08 14:16:49 2015"

卡哇伊406 （西安交通大学）

你好大神，能再帮我写一个吗？万分感谢呀，我们刚开这门课程，好羡慕你的，题目是在小学三年级学生中随机抽取10名学生，在学期初和学期末分别进行了两次推理能力测试，成绩分别是79.5和71分，标准差分别是9.124,9.940，相关系数为0.704，问两次测试成绩是否有显著差异。（这是一个双总体t检验）谢谢啦。。。。

求解代码：

#--------配对t检验  
n <- 10  
x1 <- 79.5  
x2 <- 71  
sd1 <- 9.124  
sd2 <- 9.940  
r <- 0.704  
T <- (x1-x2)/sqrt((sd1^2+sd2^2-2\*r\*sd1\*sd2) /n)  
T # 计算t统计量  
(P\_value <- 2\* (1- pt(abs(T), (2\*n-2)))) # 计算检验的p值

#---------第(1)问

Matrix <- NULL

for(i in 1:5){

Matrix <- cbind(Matrix, runif(10, 0, 10)) # 用于生成一个服从U(0,10)的均匀分布的10维向量

}

print(Matrix) # 输出生成的矩阵

#---------第(2)问

colmean1 <- function(x) colMeans(x) #这是写一个计算矩阵列均值的函数的方法1

colmean2 <- function(x) apply(x, 2, mean) #这是方法2

colmean1(Matrix)

colmean2(Matrix)

#---------第(3)问

ExtractEvenRow <- function(x) {

n <- dim(x)[1]

Result <- NULL

for(i in 1:n){

if(i %%2 ==0){ #行标为偶数，则提取出来

Result <- rbind(Result, x[i,])

}

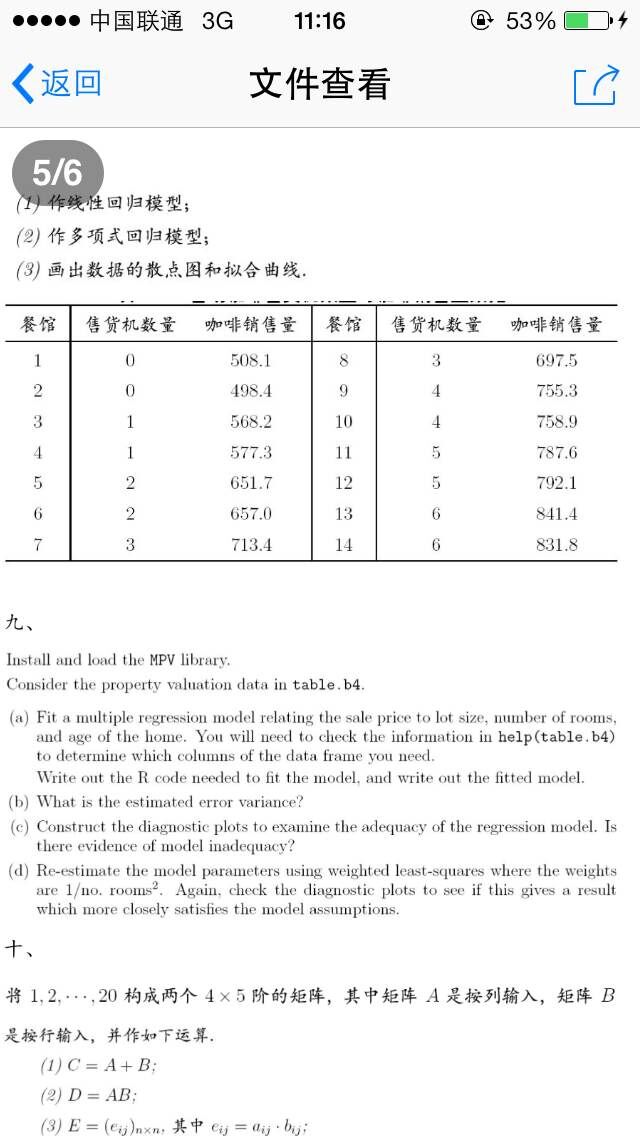
}

return(Result)

}

ExtractEvenRow(Matrix)

"Wed Jun 10 14:52:52 2015"



#------load data（数据载入）

MaNum<- c(0,0,1,1,2,2,3,3,4,4,5,5,6,6) # Machine Number

Sales <- c(508.1,498.4,568.2,577.3,651.7,657.0,713.4,697.5,755.3,758.9,787.6,792.1,841.4,831.8)

Data <- data.frame(MaNum, Sales)

#------ linear regression(线性回归)

fit1 <- lm(Sales ~ MaNum, data =Data)

summary(fit1) # linear regression is good

#------ polynomial regression(多项式回归)

fit2 <- lm(Sales ~ MaNum + I(MaNum^2)) # polynomial regression with order two(二次多项式回归)

summary(fit2) # fit information （查看拟合信息），效果也不错

fit3 <- lm(Sales ~ MaNum + I(MaNum^2)+ I(MaNum^3)) # polynomial regression with order three(三次多项式回归)

summary(fit3) # 3次时，它的高次项变量的显著性检验通不过

#------scatter plot and fit curve

par(mfrow=c(2,2))

plot(Data, main='raw data') # scatter

plot(MaNum,fitted(fit1), type='l',main='linear regression')

plot(MaNum,fitted(fit2), type='l', main='polynomial regression with order two')

plot(MaNum,fitted(fit3), type='l', main='polynomial regression with order three')

#------九 install and load the MPV library

install.packages('MPV')

library(MPV)

#-------load data

?table.b4 # make figure of the data 等价于

# help(table.b4)

str(table.b4)

head(table.b4)

#--------(a) multiple regression model

fit91 <- lm(y ~ x3+x6+x8, data= table.b4)

summary(fit91)

#--------(b) estamated error variance

var(resid(fit91))

#--------(c) diagnostic plots

par(mfrow=c(2,2))

plot(fit91)

#--------(d) Re\_estimate the model parameters

fit92 <- lm(y~ x3+x6+x8, data=table.b4, weights=1/x6^2)

summary(fit92)

win.graph()

par(mfrow=c(2,2))

plot(fit92)

"Wed Jun 10 14:55:23 2015"

#---------第(1)问

Matrix <- NULL

for(i in 1:5){

Matrix <- cbind(Matrix, runif(10, 0, 10)) # 用于生成一个服从U(0,10)的均匀分布的10维向量

}

print(Matrix) # 输出生成的矩阵

#---------第(2)问

colmean1 <- function(x) colMeans(x) #这是写一个计算矩阵列均值的函数的方法1

colmean2 <- function(x) apply(x, 2, mean) #这是方法2

colmean1(Matrix)

colmean2(Matrix)

#---------第(3)问

ExtractEvenRow <- function(x) {

n <- dim(x)[1]

Result <- NULL

for(i in 1:n){

if(i %%2 ==0){ #行标为偶数，则提取出来

Result <- rbind(Result, x[i,])

}

}

return(Result)

}

ExtractEvenRow(Matrix)