**Week3:**

str(airquality)#查看数据框结构

require(tidyverse)

airdata<-airquality%>%select(Day,Month,Temp,Wind)#用select()选列

airdata57<-airquality%>%select(Day,Month,Temp,Wind)%>%filter(Month==5|Month==6|Month==7)#用filter()选行

airdata57<-airquality%>%select(Day,Month,Temp,Wind)%>%filter(Month!=8|Month!=9)

airdata69<-airquality%>%select(-Temp)%>%filter(Month!=5)#选除了5月以外的其他行

#合并（2种合并方式）+按列合并

#full\_join()函数进行合并，根据共享的列名进行合并，保留两个数据框中的所有行

#即使某行在一个数据框中不在另一个数据框，该函数也会将它们包含在结果中

airmerge<-full\_join(airdata57,airdata69)

#merge()函数合并，all=true保留两个数据框中的所有行

airmerge2<-merge(airdata57,airdata69,all=TRUE)

airmerge2%>%arrange(Month)#按月进行排序

set.seed(0)#生成随机数

dataleft<-data.frame(id=c(1:10),height=rnorm(10,170,5))#id列是从1到10，height列的值是从均值为170，标准差为5的正态分布种生成

#rep()生成5到8每个数值重复两次

#sample()从英文字母里随机抽8个

dataright<-data.frame(id=rep(5:8,each=2),alpha=sample(letters,8))

#inner\_join（）合并只保留共有行

inner\_join(dataleft,dataright)

#left\_join()根据共有的列名进行合并，并且保留左边数据框的列名

left\_join(dataleft,dataright)

#按行合并

dataleft<-dataleft%>%mutate("weight"=rnorm(10,60,3))#mutate()添加新列

bind\_rows(dataleft,dataright)#按行合并，包含所有行

rbind(dataleft,dataright)#根据所有共有的列名进行合并，保留所有行

bind\_rows(dataleft,dataright)#将所有的数据框按列合并，生成更宽的数据

dat2<-read.csv("Set1\_emp2.csv",skip=2,header=TRUE,strip.white = TRUE)

names(dat2)[6]<-"salary(USD)"#改列名

dat1<-dat1%>%unite("name",first.name:last.name)#合并其中的两列

dat1$name<-dat1$name%>%str\_replace("\_"," ")#将这一列的下划线都换成空格

fulldata[-7,]#去掉第7行

fulldata%>%filter(status=="active")#选出这一列种所有为active值得数

#######data transformation

#######min-max normalization

set.seed(0)

data<-cbind(rnorm(1000,3,30),rnorm(1000,4,2))

minmax<-function(x){

out<-(x-min(x))/(max(x)-min(x))

return(out)

}

transformdata<-cbind(minmax(data[,1]),minmax(data[,2]))

######

###z-score

set.seed(0)

data<-cbind(rnorm(1000,3,30),rnorm(1000,4,2))

zscore<-function(x){

out<-(x-min(x))/(max(x)-min(x))

return(out)

}

transformdata<-cbind(zscore(data[,1]),zscore(data[,2]))

transformdata2<-apply(data,2,zscore)

transformdata3<-scale(data)

plot(transformdata)

plot(data)

#####

set.seed(0)

data<-cbind(rnorm(1000,3,30),rnorm(1000,4,2))

zscore<-function(x){

maxx<-max(abs(x))

j<-floor(log10(maxx))+1

out<-x/(10^j)

return(out)

}

log10(500)

10^2.69897

floor(log10(500)+1)

######log10

transformdata<-apply(data,2,decimalscale)

summary(transformdata)

summary(data)

floor(3.5)

ceiling(3.5)

#########square root/log/inverse transformation(week2)

data<-read.csv("Set1\_trans.csv")

hist(data[,1])#画出某一列的直方图

summary(data)#每一列的最小值，第一四分位数，中位数，均值，第三四分位数和最大值

hist(log(data[,1]+1))#第一列的对数值后加1 的分布

hist(sqrt(data[,3]))#第三列平方根

rbintrans<-function(x,k=3/8){

n<-length(x)

xr<-rank(x)

out<-qnorm((xr-k)/(n-2\*k+1))

return(out)

}

transformdata<-apply(data,2,rbintrans)

plot(transformdata)

plot(data)

data<-read.csv("Set1\_trans.csv")

hist(rbintrans(data[,2]))

hist(data[,1])

########box-cox

x<-data[,1]+1

out<-MASS::boxcox(lm(x~1))

lambda<-out$x[which.max(out$y)]

boxcox<-function(x){

out<-MASS::boxcox(lm(x~1))

lambda<-out$x[which.max(out$y)]

if(lambda==0){output<-log(x)}

else{output<-(x^lambda-1)/lambda}

return(output)

}

transformdata<-apply(data,2,boxcox)

plot(transformdata)

plot(data)

head(transformdata)









