Week5

Jiaju Shen

Tuesday, February 24, 2015

If you are using library(gdata) and Perl start from here

Regression(Slide 5)

```
library(gdata)
bronx1<-read.xls(file.choose(),pattern="BOROUGH",stringsAsFactors=FALSE,sheet=1,
                 perl="C:/Strawberry/perl/bin/perl.exe") #I select the directory of
                                                        #my own files
bronx1<-bronx1[which(bronx1$GROSS.SQUARE.FEET!="0"
                     & bronx1$LAND.SQUARE.FEET!="0"
                     & bronx1$SALE.PRICE!="$0"),]#By using the faster way, R reads
                                                 #everything literally as strings
attach(bronx1) # If you choose to attach, leave out the "data=." in lm regression
SALE.PRICE<-sub("\\$","",SALE.PRICE)
SALE.PRICE<-as.numeric(gsub(",","", SALE.PRICE))
GROSS.SQUARE.FEET<-as.numeric(gsub(",","", GROSS.SQUARE.FEET))
LAND.SQUARE.FEET<-as.numeric(gsub(",","", LAND.SQUARE.FEET))
plot(log(GROSS.SQUARE.FEET), log(SALE.PRICE))
m1<-lm(log(SALE.PRICE)~log(GROSS.SQUARE.FEET))
summary(m1)
abline(m1,col="red",lwd=2)
plot(resid(m1))
```

Solution model 2, Figures not shown

```
m2<-lm(log(SALE.PRICE)~log(GROSS.SQUARE.FEET)+log(LAND.SQUARE.FEET)+factor(NEIGHBORHOOD))
summary(m2)
plot(resid(m2))
m2a<-lm(log(SALE.PRICE)~0+log(GROSS.SQUARE.FEET)+log(LAND.SQUARE.FEET)+factor(NEIGHBORHOOD))
summary(m2a)
plot(resid(m2a))
```

Solution model 3 and 4, Figures not shown

```
m3<-lm(log(SALE.PRICE)~0+log(GROSS.SQUARE.FEET)+log(LAND.SQUARE.FEET)
       +factor(NEIGHBORHOOD)+factor(BUILDING.CLASS.CATEGORY))
summary(m3)
plot(resid(m3))
```

```
m4<-lm(log(SALE.PRICE)~0+log(GROSS.SQUARE.FEET)+log(LAND.SQUARE.FEET)+factor(NEIGHBORHOOD)*factor(BUILD
summary(m4)
plot(resid(m4))
```

Slide 12, A complex example

```
bronx1$SALE.PRICE<-sub("\\$","",bronx1$SALE.PRICE)
bronx1$SALE.PRICE<-as.numeric(gsub(",","", bronx1$SALE.PRICE))
bronx1$GROSS.SQUARE.FEET<-as.numeric(gsub(",","", bronx1$GROSS.SQUARE.FEET))
bronx1$LAND.SQUARE.FEET<-as.numeric(gsub(",","", bronx1$LAND.SQUARE.FEET))
bronx1$SALE.DATE<- as.Date(gsub("[^]:digit:]]","",bronx1$SALE.DATE))
bronx1$YEAR.BUILT<- as.numeric(gsub("[^]:digit:]]","",bronx1$YEAR.BUILT))
bronx1$ZIP.CODE<- as.character(gsub("[^]:digit:]]","",bronx1$ZIP.CODE))
minprice<-10000
bronx1<-bronx1[which(bronx1$SALE.PRICE>=minprice),]
nval<-dim(bronx1)[1]
bronx1$ADDRESSONLY<- gsub("[,][[:print:]]*","",gsub("[ ]+","",trim(bronx1$ADDRESS)))
bronxadd<-unique(data.frame(bronx1$ADDRESSONLY, bronx1$ZIP.CODE,stringsAsFactors=FALSE))
names(bronxadd)<-c("ADDRESSONLY","ZIP.CODE")
bronxadd<-bronxadd[order(bronxadd$ADDRESSONLY),]
duplicates<-duplicated(bronx1$ADDRESSONLY)
for(i in 1:2345) {
  if(duplicates[i]==FALSE) dupadd<-bronxadd[bronxadd$duplicates,1]
}#Did not quite understand what we are doing with dupadd. Wrong one.
nsample=450
addsample<-bronxadd[sample.int(dim(bronxadd),size=nsample),]#I use nval here
library(ggmap)
addrlist<-paste(addsample$ADDRESSONLY, "NY", addsample$ZIP.CODE, "US", sep=" ")
querylist<-geocode(addrlist) #This is cool. Take a break.
matched<-(querylist$lat!=0 &&querylist$lon!=0)
addsample<-cbind(addsample,querylist$lat,querylist$lon)
names(addsample)<-c("ADDRESSONLY","ZIPCODE","Latitude","Longitude")# correct the column names. Worked
adduse<-merge(bronx1,addsample)
adduse<-adduse[!is.na(adduse$Latitude),]
mapcoord<-adduse[,c(2,3,24,25)]
table(mapcoord$NEIGHBORHOOD)
mapcoord$NEIGHBORHOOD <- as.factor(mapcoord$NEIGHBORHOOD)
map <- get_map(location = 'Bronx', zoom = 12)#Zoom 11 or 12
ggmap(map) + geom_point(aes(x = mapcoord$Longitude, y = mapcoord$Latitude, size =1,
                            color=mapcoord$NEIGHBORHOOD), data = mapcoord)
                            +theme(legend.position = "none")
#It would be perfect if I can decrease the size of points
mapmeans<-cbind(adduse,as.numeric(mapcoord$NEIGHBORHOOD))
colnames(mapmeans)[26] <- "NEIGHBORHOOD" #This is the right way of renaming.
keeps <- c("ZIP.CODE","NEIGHBORHOOD","TOTAL.UNITS","LAND.SQUARE.FEET","GROSS.SQUARE.FEET",
           "SALE.PRICE","Latitude","Longitude")
mapmeans<-mapmeans[keeps]#Dropping others
mapmeans$NEIGHBORHOOD<-as.numeric(mapcoord$NEIGHBORHOOD)
for(i in 1:8){
  mapmeans[,i]=as.numeric(mapmeans[,i])
}#Now done for conversion to numeric
```

```
mapobj<-kmeans(mapmeans,5, iter.max=10, nstart=5, algorithm = c("Hartigan-Wong", "Lloyd",
                                                                "Forgy", "MacQueen"))
fitted(mapobj,method=c("centers","classes"))
mapobj$centers
library(cluster)
clusplot(mapmeans, mapobj$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
library(fpc)#Need to install.packages("fpc")
plotcluster(mapmeans, mapobj$cluster)
mapmeans1<-mapmeans[,-c(1,3,4)]
mapobjnew<-kmeans(mapmeans1,5, iter.max=10, nstart=5, algorithm = c("Hartigan-Wong", "Lloyd",
                                                                    "Forgy", "MacQueen"))
fitted(mapobjnew,method=c("centers","classes"))
clusplot(mapmeans1, mapobjnew$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
plotcluster(mapmeans1, mapobjnew$cluster)
ggmap(map) + geom_point(aes(x = mapcoord$Longitude, y = mapcoord$Latitude, size =1,
                            color=mapobjnew$cluster), data = mapcoord)#How to change colors?
```