Homework 2

Roozbeh

7/21/2019

#read the googleplay3.csv file and place it in a data frame called df.  
df<-read.csv("C:/Users/10/Documents/Data Science/Data Science Course - Yaser Zerehsaz/HW/googleplaystore3-1.csv", header=TRUE, stringsAsFactors = F)   
  
#Get the summary and structure of df.  
summary(df)

## X App Category Rating   
## Min. : 0 Length:10839 Length:10839 Min. :1.000   
## 1st Qu.: 2710 Class :character Class :character 1st Qu.:4.000   
## Median : 5419 Mode :character Mode :character Median :4.300   
## Mean : 5419 Mean :4.192   
## 3rd Qu.: 8128 3rd Qu.:4.500   
## Max. :10840 Max. :5.000   
## NA's :1473   
## Reviews Type Last.Updated   
## Min. : 0 Length:10839 Length:10839   
## 1st Qu.: 37 Class :character Class :character   
## Median : 2062 Mode :character Mode :character   
## Mean : 447611   
## 3rd Qu.: 54048   
## Max. :78158306   
## NA's :200   
## Installs   
## Length:10839   
## Class :character   
## Mode :character   
##   
##   
##   
##

str(df)

## 'data.frame': 10839 obs. of 8 variables:  
## $ X : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ App : chr "Photo Editor & Candy Camera & Grid & ScrapBook" "Coloring book moana" "U Launcher Lite â\200“ FREE Live Cool Themes, Hide Apps" "Sketch - Draw & Paint" ...  
## $ Category : chr "ART\_AND\_DESIGN" "ART\_AND\_DESIGN" "ART\_AND\_DESIGN" "ART\_AND\_DESIGN" ...  
## $ Rating : num 4.1 3.9 4.7 4.5 4.3 4.4 3.8 4.1 4.4 4.7 ...  
## $ Reviews : int 159 967 87510 215644 967 167 178 36815 13791 121 ...  
## $ Type : chr "Free" "Free" "Free" "Free" ...  
## $ Last.Updated: chr "1/7/2018" "1/15/2018" "8/1/2018" "6/8/2018" ...  
## $ Installs : chr "10,000+" "500,000+" "5,000,000+" "50,000,000+" ...

#Are there any missing values in the data? Which variables?  
apply(is.na(df), 2, sum)

## X App Category Rating Reviews   
## 0 0 0 1473 200   
## Type Last.Updated Installs   
## 0 0 0

#Change the levels of variable “Category” to 1 to 33.  
df$Category=factor(df$Category,  
 levels = levels(as.factor(df$Category)),  
 labels = seq(length(levels(as.factor(df$Category))))  
   
 )  
  
#pply unique function to the variable “Type” and comment on its unique values.  
unique(df$Type)

## [1] "Free" " " "Paid"

sum(is.na(df$type))

## [1] 0

sum(df$Type == " ")

## [1] 400

#It has no NA so it hasn't any missing values but it has 400 empty values.  
  
# Treating Missing Values  
  
#Replace the empty values with NA  
df$Type[df$Type == " "]= NA  
  
#Get the levels of variable “Type” using the levels function.   
levels(as.factor(df$Type))

## [1] "Free" "Paid"

#Get the summary of df  
summary(df)

## X App Category Rating   
## Min. : 0 Length:10839 12 :1971 Min. :1.000   
## 1st Qu.: 2710 Class :character 15 :1144 1st Qu.:4.000   
## Median : 5419 Mode :character 30 : 843 Median :4.300   
## Mean : 5419 21 : 463 Mean :4.192   
## 3rd Qu.: 8128 5 : 460 3rd Qu.:4.500   
## Max. :10840 26 : 424 Max. :5.000   
## (Other):5534 NA's :1473   
## Reviews Type Last.Updated   
## Min. : 0 Length:10839 Length:10839   
## 1st Qu.: 37 Class :character Class :character   
## Median : 2062 Mode :character Mode :character   
## Mean : 447611   
## 3rd Qu.: 54048   
## Max. :78158306   
## NA's :200   
## Installs   
## Length:10839   
## Class :character   
## Mode :character   
##   
##   
##   
##

#Check the NA’s for the variable “Type”  
apply(is.na(df), 2, sum)

## X App Category Rating Reviews   
## 0 0 0 1473 200   
## Type Last.Updated Installs   
## 400 0 0

#The “Last.Updated” column should be in “Date” format. Is it? If not, we need to coerce it to be a date variable so that R knows these are dates and not characters.  
df$Last.Updated<-as.Date(df$Last.Updated,   
 tryFormats =c("%m/%d/%Y","%Y/%d/%m","%Y-%d-%m","%m-%d-%Y","%d/%m/%Y","%Y/%m/%d","%Y-%m-%d","%d-%m-%Y")  
 )  
  
#we need to compute a new variable called “Updates”. The new variable should give the number of days since the last update of the application  
df$Updates =as.numeric(Sys.Date()-df$Last.Updated)  
  
#Get the str(df) and check the structures.  
str(df)

## 'data.frame': 10839 obs. of 9 variables:  
## $ X : int 0 1 2 3 4 5 6 7 8 9 ...  
## $ App : chr "Photo Editor & Candy Camera & Grid & ScrapBook" "Coloring book moana" "U Launcher Lite â\200“ FREE Live Cool Themes, Hide Apps" "Sketch - Draw & Paint" ...  
## $ Category : Factor w/ 33 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Rating : num 4.1 3.9 4.7 4.5 4.3 4.4 3.8 4.1 4.4 4.7 ...  
## $ Reviews : int 159 967 87510 215644 967 167 178 36815 13791 121 ...  
## $ Type : chr "Free" "Free" "Free" "Free" ...  
## $ Last.Updated: Date, format: "2018-01-07" "2018-01-15" ...  
## $ Installs : chr "10,000+" "500,000+" "5,000,000+" "50,000,000+" ...  
## $ Updates : num 561 553 355 409 397 848 452 403 670 384 ...

#Get the first five values of the variable “Installs” in the dataset.  
head(df$Installs,5)

## [1] "10,000+" "500,000+" "5,000,000+" "50,000,000+" "100,000+"

#We need to remove the “+” sign from the end of the number of installs and change the whole column into a numeric variable.  
makenumeric = function(x){  
 spinst = unlist(strsplit(x,''))  
 spinst=spinst[spinst!=',']  
 spinst = paste(spinst[-c(length(spinst))],collapse ='')  
 ;return (spinst)  
 }  
  
for (i in 1:length(df$Installs)){  
 df$Installs[i] = makenumeric(df$Installs[i])  
}  
  
#remove the first, second, seventh and eighth columns from the dataset.  
keeps = c("Category","Rating","Reviews","Type","Updates")  
d=df[keeps]  
  
  
  
#Use the margin and aggregations plots to interpret the missing values patterns.  
library(mice) ;echo=FALSE

## Warning: package 'mice' was built under R version 3.6.1

## Loading required package: lattice

##   
## Attaching package: 'mice'

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(VIM)

## Loading required package: colorspace

## Loading required package: grid

## Loading required package: data.table

## Registered S3 methods overwritten by 'car':  
## method from  
## influence.merMod lme4  
## cooks.distance.influence.merMod lme4  
## dfbeta.influence.merMod lme4  
## dfbetas.influence.merMod lme4

## VIM is ready to use.   
## Since version 4.0.0 the GUI is in its own package VIMGUI.  
##   
## Please use the package to use the new (and old) GUI.

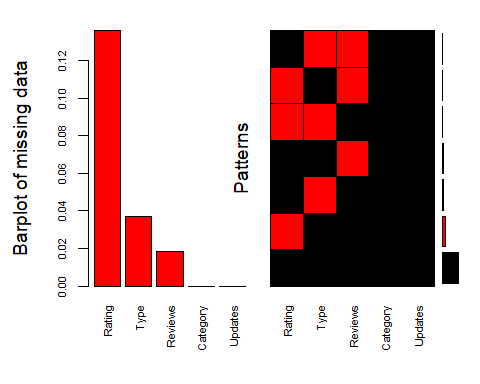
## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues

##   
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':  
##   
## sleep

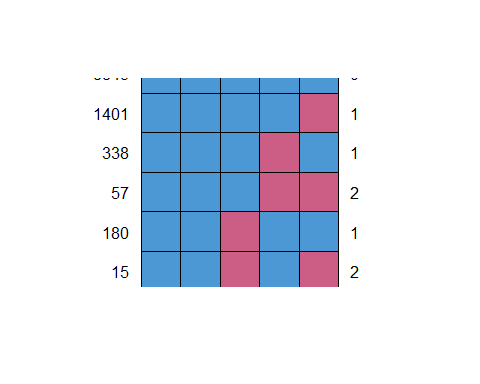
aggr<- aggr(d, col=c('black','red'), numbers=TRUE, sortVars=TRUE,   
labels=names(d), cex.axis=.7, gap=1, ylab=c("Barplot of missing data","Patterns"))

## Warning in plot.aggr(res, ...): not enough horizontal space to display  
## frequencies



##   
## Variables sorted by number of missings:   
## Variable Count  
## Rating 0.13589815  
## Type 0.03690377  
## Reviews 0.01845189  
## Category 0.00000000  
## Updates 0.00000000

md.pattern(d)

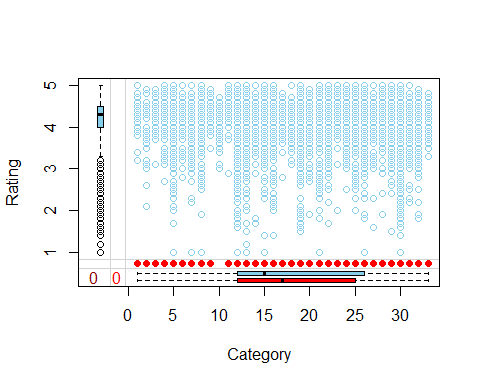
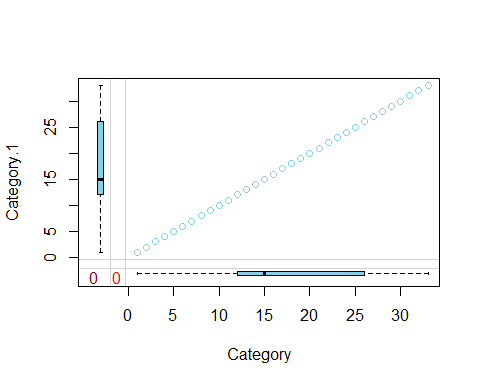


## Category Updates Reviews Type Rating   
## 8843 1 1 1 1 1 0  
## 1401 1 1 1 1 0 1  
## 338 1 1 1 0 1 1  
## 57 1 1 1 0 0 2  
## 180 1 1 0 1 1 1  
## 15 1 1 0 1 0 2  
## 5 1 1 0 0 1 2  
## 0 0 200 400 1473 2073

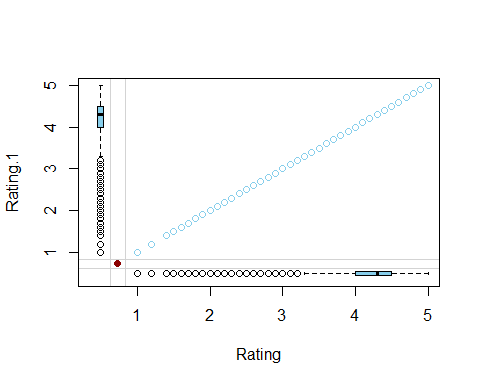
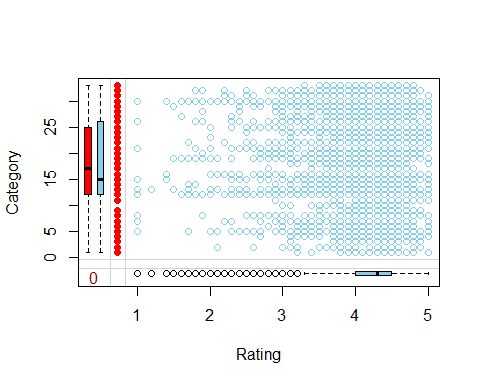
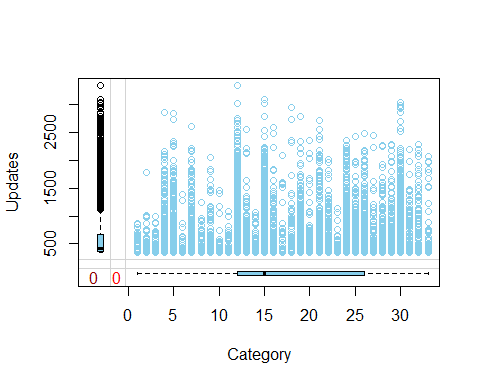
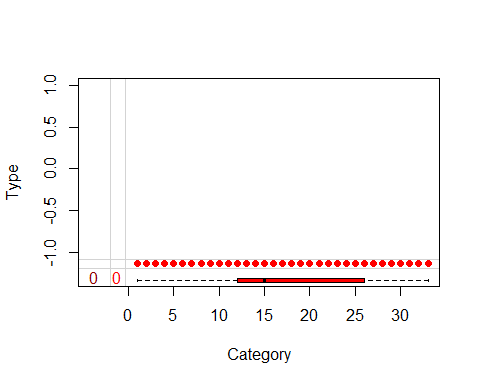
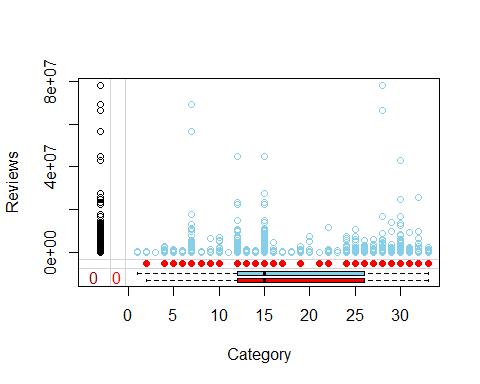
md.pairs(d)

## $rr  
## Category Rating Reviews Type Updates  
## Category 10839 9366 10639 10439 10839  
## Rating 9366 9366 9181 9023 9366  
## Reviews 10639 9181 10639 10244 10639  
## Type 10439 9023 10244 10439 10439  
## Updates 10839 9366 10639 10439 10839  
##   
## $rm  
## Category Rating Reviews Type Updates  
## Category 0 1473 200 400 0  
## Rating 0 0 185 343 0  
## Reviews 0 1458 0 395 0  
## Type 0 1416 195 0 0  
## Updates 0 1473 200 400 0  
##   
## $mr  
## Category Rating Reviews Type Updates  
## Category 0 0 0 0 0  
## Rating 1473 0 1458 1416 1473  
## Reviews 200 185 0 195 200  
## Type 400 343 395 0 400  
## Updates 0 0 0 0 0  
##   
## $mm  
## Category Rating Reviews Type Updates  
## Category 0 0 0 0 0  
## Rating 0 1473 15 57 0  
## Reviews 0 15 200 5 0  
## Type 0 57 5 400 0  
## Updates 0 0 0 0 0

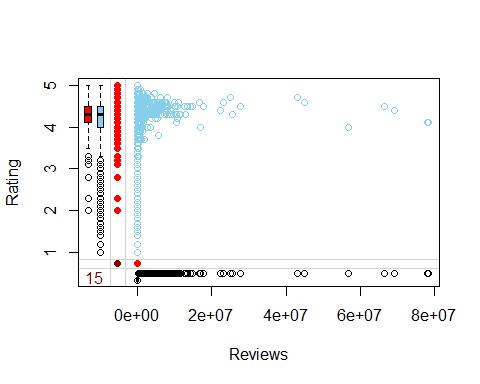
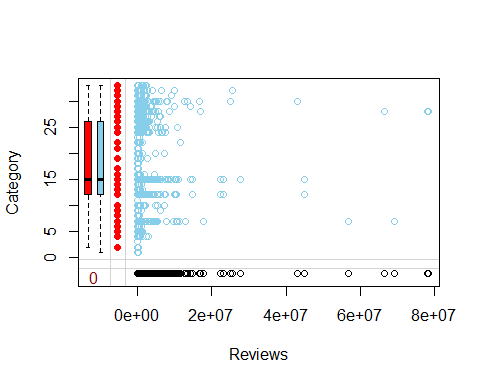
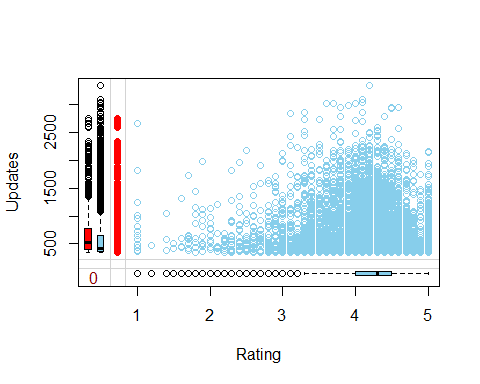
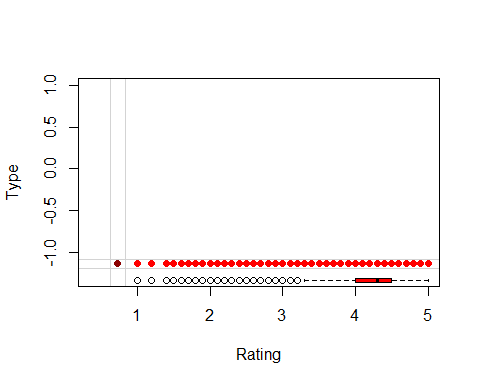
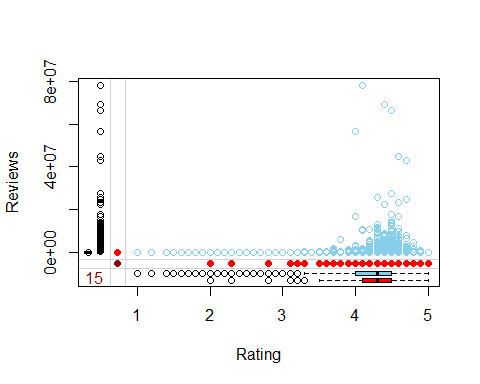
for (i in seq(5)){  
 for (k in seq(5)){  
 marginplot(d[,c(i,k)])  
 }  
}



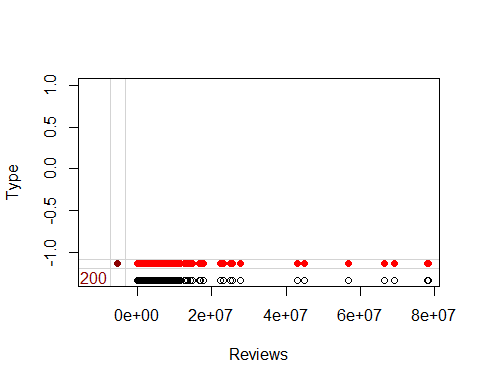
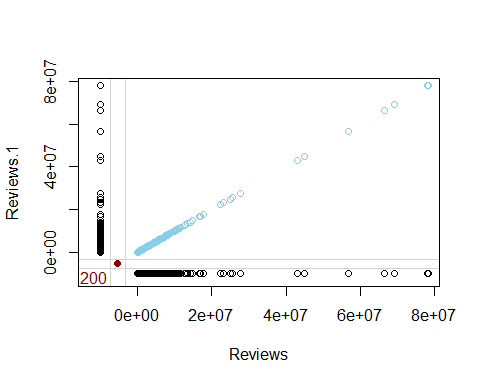
## Warning in data.matrix(x): NAs introduced by coercion



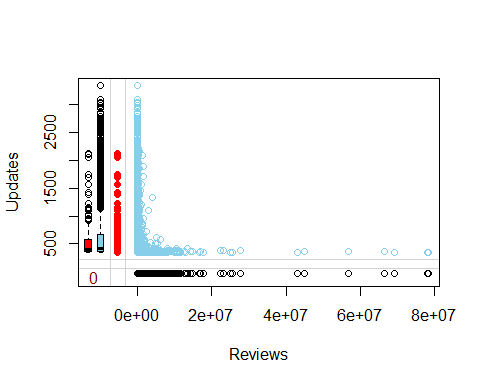
## Warning in data.matrix(x): NAs introduced by coercion



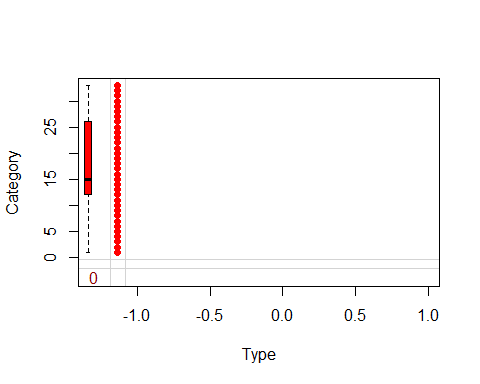
## Warning in data.matrix(x): NAs introduced by coercion



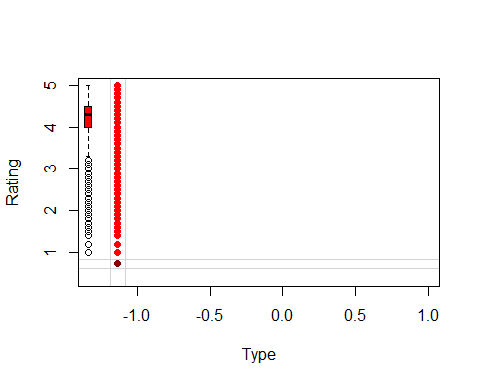
## Warning in data.matrix(x): NAs introduced by coercion



## Warning in data.matrix(x): NAs introduced by coercion

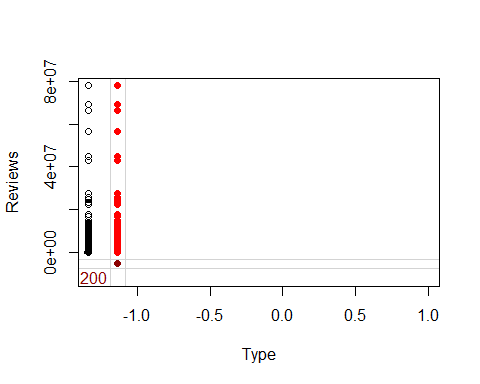


## Warning in data.matrix(x): NAs introduced by coercion

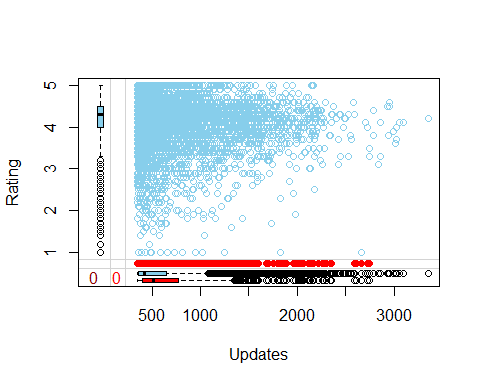
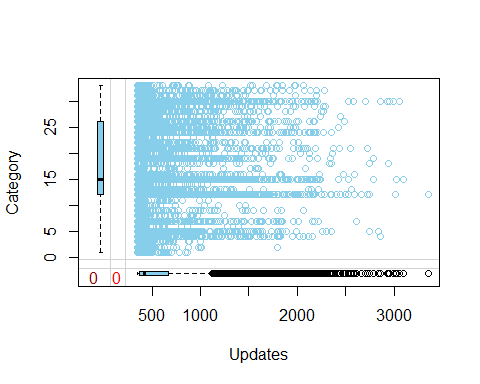
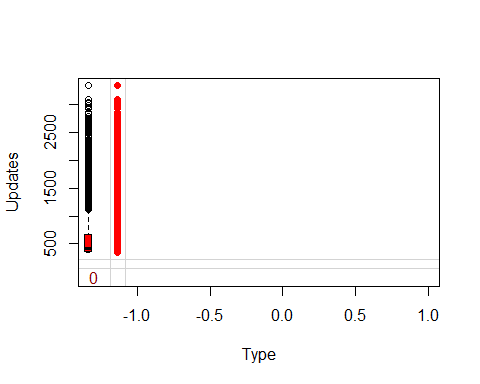
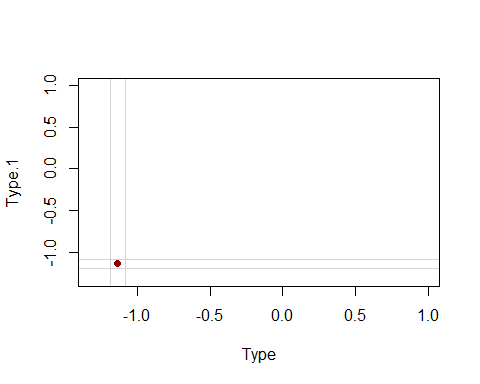


## Warning in data.matrix(x): NAs introduced by coercion

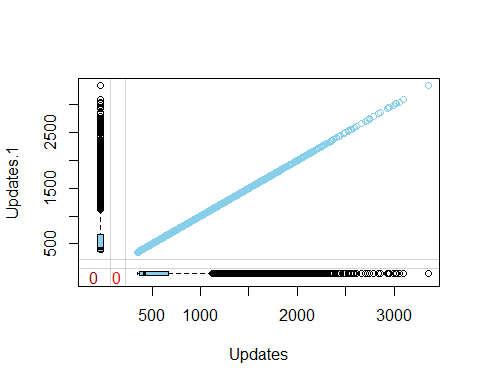
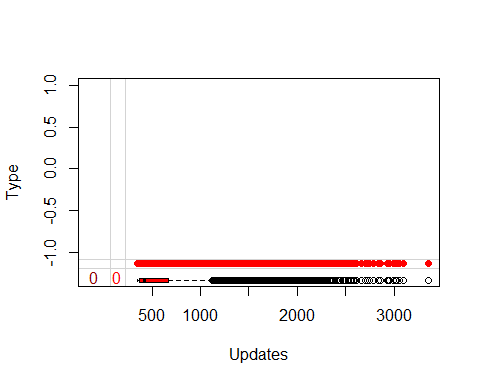
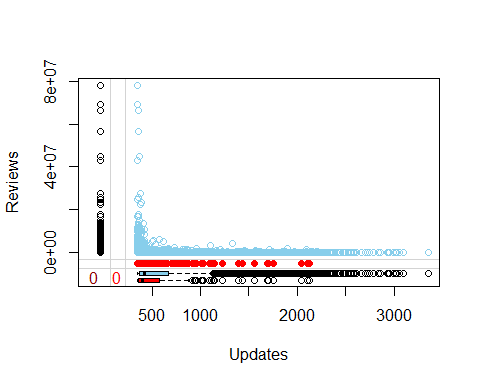
## Warning in data.matrix(x): NAs introduced by coercion



## Warning in data.matrix(x): NAs introduced by coercion



## Warning in data.matrix(x): NAs introduced by coercion

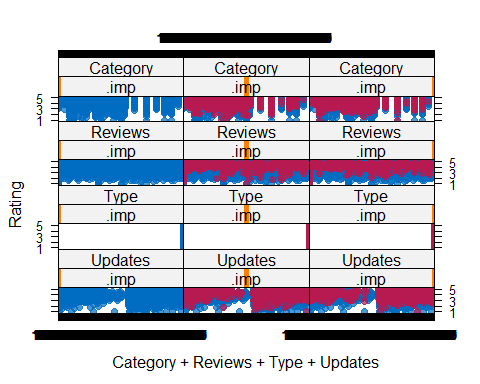


#Use the function mice with m = 2 to impute the missing values.  
imput<-mice(d,m=2,meth=c("polyreg","sample","pmm","logreg","pmm"))

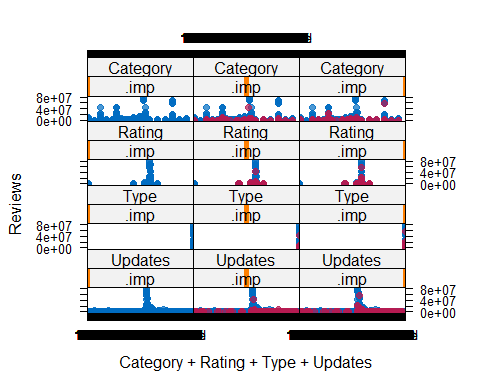
##   
## iter imp variable  
## 1 1 Rating Reviews  
## 1 2 Rating Reviews  
## 2 1 Rating Reviews  
## 2 2 Rating Reviews  
## 3 1 Rating Reviews  
## 3 2 Rating Reviews  
## 4 1 Rating Reviews  
## 4 2 Rating Reviews  
## 5 1 Rating Reviews  
## 5 2 Rating Reviews

## Warning: Number of logged events: 1

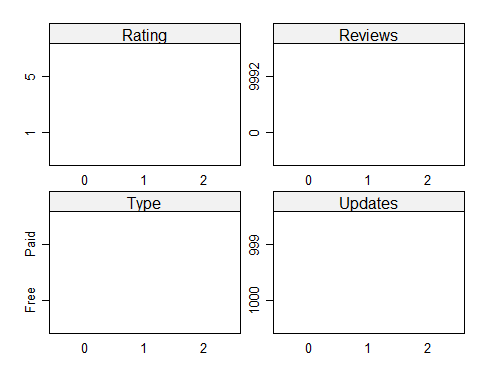
#Use the xyplot, stripplot and densityplot to comment on the performance of imputation.  
xyplot(imput, Rating ~ Category+Reviews+Type+Updates|.imp, pch = 20, cex = 1.4)



xyplot(imput, Reviews ~ Category+Rating+Type+Updates|.imp, pch = 20, cex = 1.4)



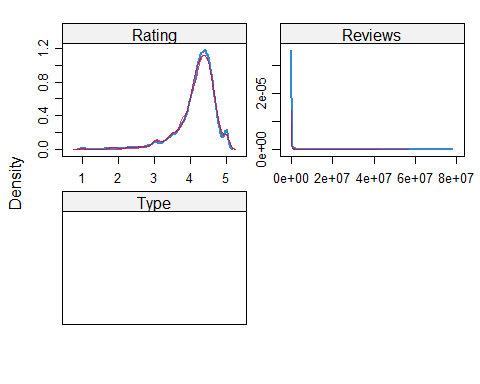
stripplot(imput, pch = 20, cex = 1.2)



densityplot(imput)

## Warning in (function (x, darg, groups = NULL, weights = NULL, subscripts =  
## TRUE, : NAs introduced by coercion

## Warning in panel.superpose(x, darg = darg, plot.points = plot.points, ref =  
## FALSE, : NAs introduced by coercion



#Perform the imputation using the complete function and get the summary of the resulting data frames.  
com<- complete(imput,1)   
  
summary(com)

## Category Rating Reviews Type   
## 12 :1971 Min. :1.00 Min. : 0 Length:10839   
## 15 :1144 1st Qu.:4.00 1st Qu.: 36 Class :character   
## 30 : 843 Median :4.30 Median : 2046 Mode :character   
## 21 : 463 Mean :4.19 Mean : 447917   
## 5 : 460 3rd Qu.:4.50 3rd Qu.: 53745   
## 26 : 424 Max. :5.00 Max. :78158306   
## (Other):5534   
## Updates   
## Min. : 348.0   
## 1st Qu.: 367.0   
## Median : 424.0   
## Mean : 607.7   
## 3rd Qu.: 670.0   
## Max. :3349.0   
##