LasVegas

Andie

9/1/2017

```
#install.packages('corrplot')
#install.packages('doBy')
library(doBy) #summaryBy
library(ggplot2) # Data visualization
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(corrplot) # correlation plot
library(plotly) # interactive map

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout

library(rworldmap) #world map

## Loading required package: sp

## ### Welcome to rworldmap ###

## For a short introduction type :   vignette('rworldmap')
```

Lets open our file. We notice that its a csv file and it uses semi colons to separate the values, so we will use read.csv2 to open our file and assign it to 'the 'df'.

```
df <- read.csv2('/Users/andiedonovan/myProjects/LasVegas/LasVegas.csv') # open file
#View(df) # view the whole data set
```

We notice that some of the column names are long or messy, so lets fix that:

There are a lot of columns, so it takes a long time to rename the columns. Say we didnt really care about their length, but just wanted to remove the periods from the column names. We could use the function gsub to easily do this:

```
names(df) <- gsub("\\.", "", names(df))
head(df)

##   Usercountry Nrreviews Nrhotelreviews Helpfulvotes Score Periodofstay
## 1         USA        11              4           13     5      Dec-Feb
## 2         USA       119             21           75     3      Dec-Feb
## 3         USA        36              9           25     5      Mar-May
## 4          UK        14              7           14     4      Mar-May
## 5      Canada         5              5            2     4      Mar-May
## 6      Canada        31              8           27     3      Mar-May
##   Travelertype Pool Gym Tenniscourt Spa Casino Freeinternet
## 1      Friends   NO YES          NO  NO    YES          YES
## 2     Business   NO YES          NO  NO    YES          YES
## 3     Families   NO YES          NO  NO    YES          YES
## 4      Friends   NO YES          NO  NO    YES          YES
## 5         Solo   NO YES          NO  NO    YES          YES
## 6      Couples   NO YES          NO  NO    YES          YES
##                               Hotelname Hotelstars Nrrooms Usercontinent
## 1 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 2 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 3 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 4 Circus Circus Hotel & Casino Las Vegas          3    3773        Europe
## 5 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 6 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
##   Memberyears Reviewmonth Reviewweekday
## 1           9     January      Thursday
## 2           3     January        Friday
## 3           2    February      Saturday
## 4           6    February        Friday
## 5           7       March       Tuesday
## 6           2       March       Tuesday
```

If we did want to go ahead and rename all of the columns, we could do that using names, colnames and a list:

```
#Using names
names(df) # our current column names

##  [1] "Usercountry"    "Nrreviews"      "Nrhotelreviews" "Helpfulvotes"
##  [5] "Score"          "Periodofstay"   "Travelertype"   "Pool"
##  [9] "Gym"            "Tenniscourt"    "Spa"            "Casino"
## [13] "Freeinternet"   "Hotelname"      "Hotelstars"     "Nrrooms"
## [17] "Usercontinent"  "Memberyears"    "Reviewmonth"    "Reviewweekday"

names(df)[1]<-"Country" #rename just first one column
head(df) #check that the first column name changed from 'Usercountry' to 'Country'

##   Country Nrreviews Nrhotelreviews Helpfulvotes Score Periodofstay
## 1     USA        11              4           13     5      Dec-Feb
## 2     USA       119             21           75     3      Dec-Feb
## 3     USA        36              9           25     5      Mar-May
## 4      UK        14              7           14     4      Mar-May
## 5  Canada         5              5            2     4      Mar-May
## 6  Canada        31              8           27     3      Mar-May
##   Travelertype Pool Gym Tenniscourt Spa Casino Freeinternet
## 1      Friends   NO YES          NO  NO    YES          YES
## 2     Business   NO YES          NO  NO    YES          YES
## 3     Families   NO YES          NO  NO    YES          YES
## 4      Friends   NO YES          NO  NO    YES          YES
## 5         Solo   NO YES          NO  NO    YES          YES
## 6      Couples   NO YES          NO  NO    YES          YES
##                                Hotelname Hotelstars Nrrooms Usercontinent
## 1 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 2 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 3 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 4 Circus Circus Hotel & Casino Las Vegas          3    3773        Europe
## 5 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
## 6 Circus Circus Hotel & Casino Las Vegas          3    3773 North America
##   Memberyears Reviewmonth Reviewweekday
## 1           9     January      Thursday
## 2           3     January        Friday
## 3           2    February      Saturday
## 4           6    February        Friday
## 5           7       March       Tuesday
## 6           2       March       Tuesday
```

```
#Using colnames
colnames(df) # gives us the same thing
```

```
##  [1] "Country"        "Nrreviews"      "Nrhotelreviews" "Helpfulvotes"
##  [5] "Score"          "Periodofstay"   "Travelertype"   "Pool"
##  [9] "Gym"            "Tenniscourt"    "Spa"            "Casino"
## [13] "Freeinternet"   "Hotelname"      "Hotelstars"     "Nrrooms"
## [17] "Usercontinent"  "Memberyears"    "Reviewmonth"    "Reviewweekday"
```

```
colnames(df)<-c('Country', 'NoReviews', 'NoHotelReviews', 'Helpful', 'Score', 'Stay',
'Traveler', 'Pool', 'Gym', 'Tennis', 'Spa', 'Casino', 'Internet', 'Name', 'Stars',
'NoRms', 'Continent', 'MemberYrs', 'Month', 'Weekday') #rename all of the columns using a
list

head(df) # check that it worked!
```

```
##   Country NoReviews NoHotelReviews Helpful Score    Stay Traveler Pool Gym
## 1     USA        11              4      13     5 Dec-Feb  Friends   NO YES
## 2     USA       119             21      75     3 Dec-Feb Business   NO YES
## 3     USA        36              9      25     5 Mar-May Families   NO YES
## 4      UK        14              7      14     4 Mar-May  Friends   NO YES
## 5  Canada         5              5       2     4 Mar-May     Solo   NO YES
## 6  Canada        31              8      27     3 Mar-May  Couples   NO YES
##   Tennis Spa Casino Internet                                   Name Stars
## 1     NO  NO    YES      YES Circus Circus Hotel & Casino Las Vegas     3
## 2     NO  NO    YES      YES Circus Circus Hotel & Casino Las Vegas     3
## 3     NO  NO    YES      YES Circus Circus Hotel & Casino Las Vegas     3
## 4     NO  NO    YES      YES Circus Circus Hotel & Casino Las Vegas     3
## 5     NO  NO    YES      YES Circus Circus Hotel & Casino Las Vegas     3
## 6     NO  NO    YES      YES Circus Circus Hotel & Casino Las Vegas     3
##   NoRms     Continent MemberYrs    Month  Weekday
## 1  3773 North America         9  January Thursday
## 2  3773 North America         3  January   Friday
## 3  3773 North America         2 February Saturday
## 4  3773        Europe         6 February   Friday
## 5  3773 North America         7    March  Tuesday
## 6  3773 North America         2    March  Tuesday
```

Theres a lot of variables in our dataset, and we probably will not need all of them for our analysis. We can create different sub datasets to allow for easier analysis of specific factors. For example, to see how whether or not a hotel has certain amenities affects its score, lets make a dataset that only includes the hotel name, country, score, and the 6 amenities listed. Lets call this new dataset amenities1:

Also notice the values for each of the variables is either 'YES' or 'NO'. Its much easier to work with numbers than characters, so lets turn each of the variables into a binary factor with 'YES'=2 and 'NO'=1

```
amenities1 = df %>%
  select(Name, Country,Score, Pool, Gym, Tennis, Spa, Casino, Internet)

amenities1$Pool<-as.numeric(amenities1$Pool)
amenities1$Gym<-as.numeric(amenities1$Gym)
amenities1$Tennis<-as.numeric(amenities1$Tennis)
amenities1$Spa<-as.numeric(amenities1$Spa)
amenities1$Casino<-as.numeric(amenities1$Casino)
amenities1$Internet<-as.numeric(amenities1$Internet)

head(amenities1)

##                                      Name Country Score Pool Gym Tennis Spa
## 1 Circus Circus Hotel & Casino Las Vegas     USA     5    1   2      1   1
## 2 Circus Circus Hotel & Casino Las Vegas     USA     3    1   2      1   1
## 3 Circus Circus Hotel & Casino Las Vegas     USA     5    1   2      1   1
## 4 Circus Circus Hotel & Casino Las Vegas      UK     4    1   2      1   1
## 5 Circus Circus Hotel & Casino Las Vegas  Canada     4    1   2      1   1
## 6 Circus Circus Hotel & Casino Las Vegas  Canada     3    1   2      1   1
##   Casino Internet
## 1      2        2
## 2      2        2
## 3      2        2
## 4      2        2
## 5      2        2
## 6      2        2
```

Above you'll see a symbol that looks like this: %>%. This is called a pipe operator (dplyr package) and is used to insert some input or argument into a function. Its useful when we have sequences of operations and can help make the flow of the steps more readable/ easier to follow.

Equivalently, we could nest the above code and get the same results. This would look like: amenities = select(df, c(Name, Country, Score, Pool, Gym, Tennis, Spa, Casino, Internet)). Although in this case, it doesnt make much of a difference which format you use, later on, the pipe operator is a powerful tool for applying multiple operations to one object.

Alternative data set without hotel name & country (ie only numerical):

```
amenities2 = df %>%
  select(Score, Pool, Gym, Tennis, Spa, Casino, Internet) # columns to be selected

amenities2 = amenities2 %>% mutate(Pool=as.factor(ifelse(Pool=="YES", "1", "0")))
amenities2 = amenities2 %>% mutate(Gym=as.factor(ifelse(Gym=="YES", "1", "0")))
```

```
amenities2 = amenities2 %>% mutate(Tennis=as.factor(ifelse(Tennis=="YES", "1", "0")))
amenities2 = amenities2 %>% mutate(Spa=as.factor(ifelse(Spa=="YES", "1", "0")))
amenities2 = amenities2 %>% mutate(Casino=as.factor(ifelse(Casino=="YES", "1", "0")))
amenities2 = amenities2 %>% mutate(Internet=as.factor(ifelse(Internet=="YES", "1", "0")))
```

Linear Regression: Lets see how influential the 6 amenities are on the hotel's score. To do this, lets regress Pool, Gym, Tennis, Spa, Casino, and Internet on Score. We will call our linear model 'am_model'

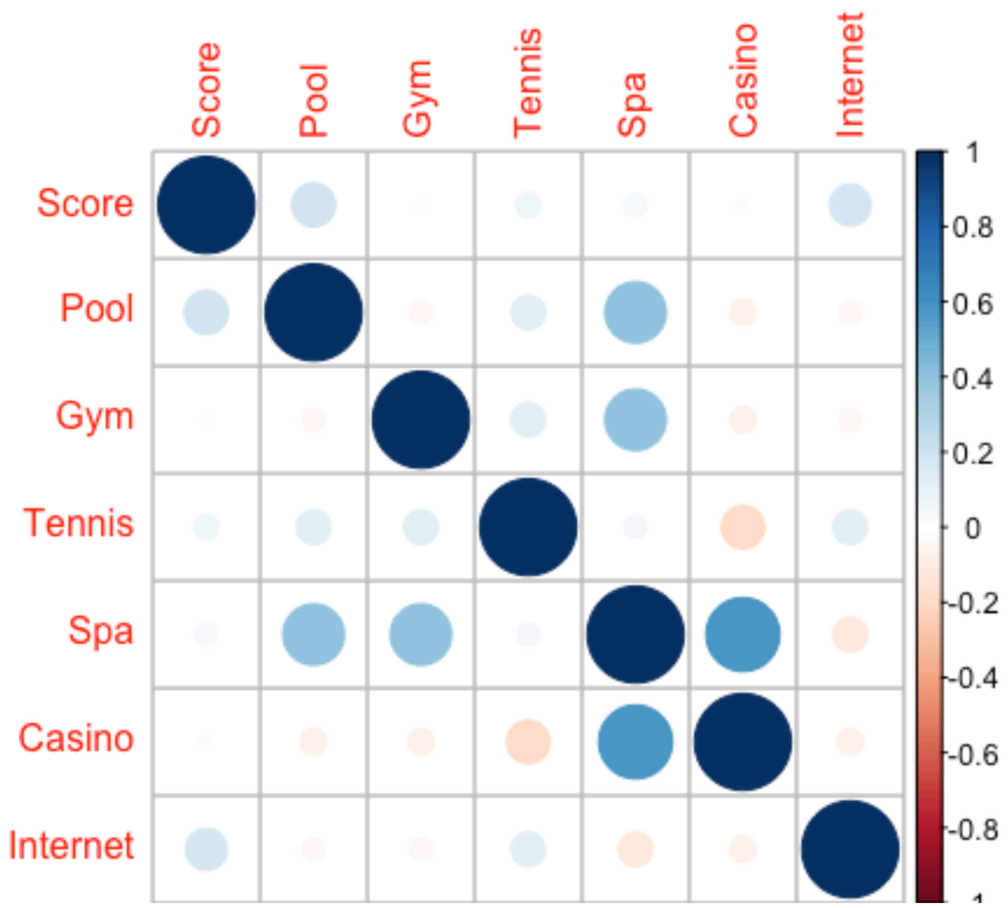
```
#am_model1<-glm(Score~Pool+Gym+Tennis+Spa+Casino+Internet, data=amenities1,
family='binomial') # run regression
#summary(am_model1) # summarize linear model

#am_model2<-glm(Score~., data=amenities2, family='binomial') # run regression
#summary(am_model2) # summarize linear model
```

The first argument in the glm (generalized linear model) function is the formula. You place the independent variable on the left of the tilda and the regressors or dependent variables on the right. The second argument simply specifies that we are using our amenities data set and the third specifies that our data is binomial (ie uses logit link function).

Look at the summary table and see if you can interpert the different columns. Recall: * Estimate (Coefficient): * Std. Error: * T Value: * P(>|t|)

Correlation Matrix: Are the variables related to each other in any way? What variables are most related to Score?

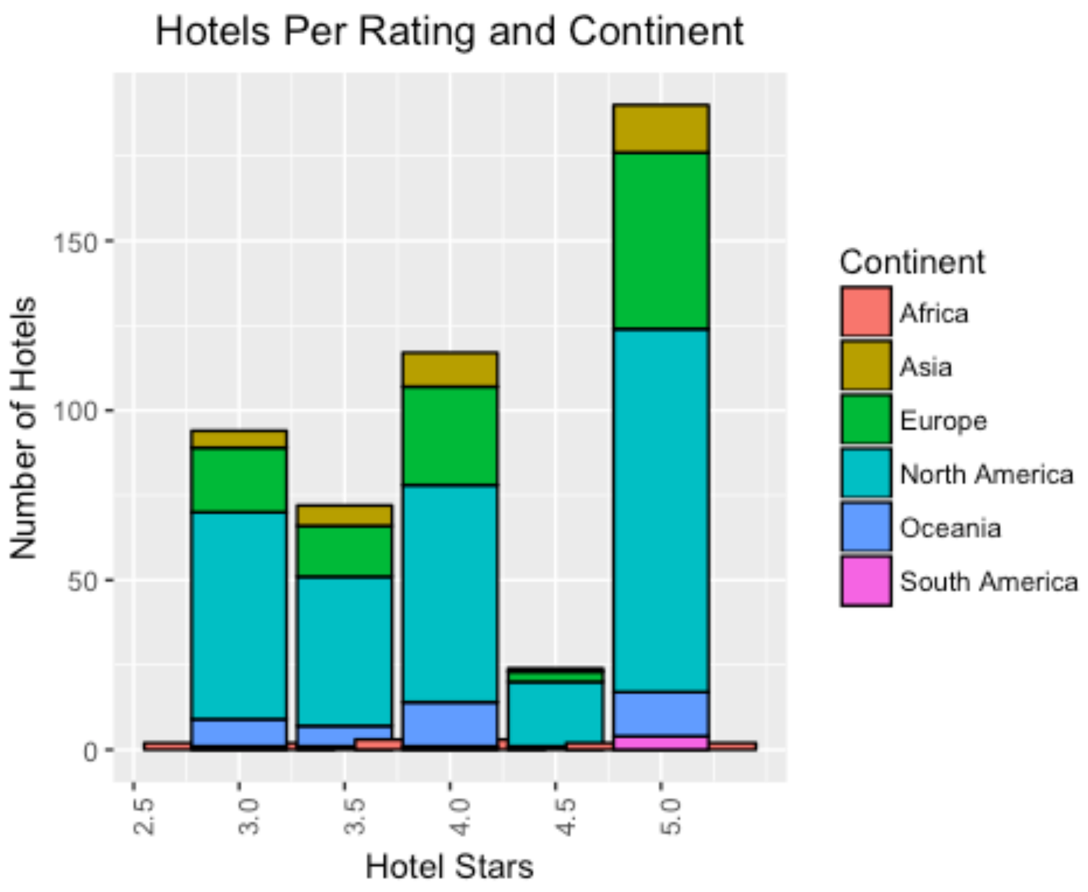
```
#amenities1[,3:9]
cor.amen<-cor(amenities1[,3:9], use="complete", method="pearson")
corrplot(cor.amen)
```



TRY: geom_bar(stat='identity')???

Lets make a graph looking at how many hotels f

```
g<- ggplot(df)+
  geom_bar(aes(x=Stars, fill = Continent), color="black") + #create bars & fill color according to continent
  theme(plot.title=element_text(hjust=0.5), # center title
        axis.text.x=element_text(angle=90, hjust=1)) + # tilt the x-axis labels 90 degrees to the left
  xlab("Hotel Stars") + # add label x axis
  ylab("Number of Hotels") + # add label y axis
  scale_x_continuous(breaks= seq(2,5,by=0.5), # make our tick marks count half stars
                     labels = c('2.0', '2.5', '3.0', '3.5','4.0', '4.5', '5.0')) +
  ggtitle("Hotels Per Rating and Continent") #title ggplot
g #shows plot

## Warning: position_stack requires non-overlapping x intervals
```



```
#ggplotly(g) # turn into plotly plot (interactive)
```