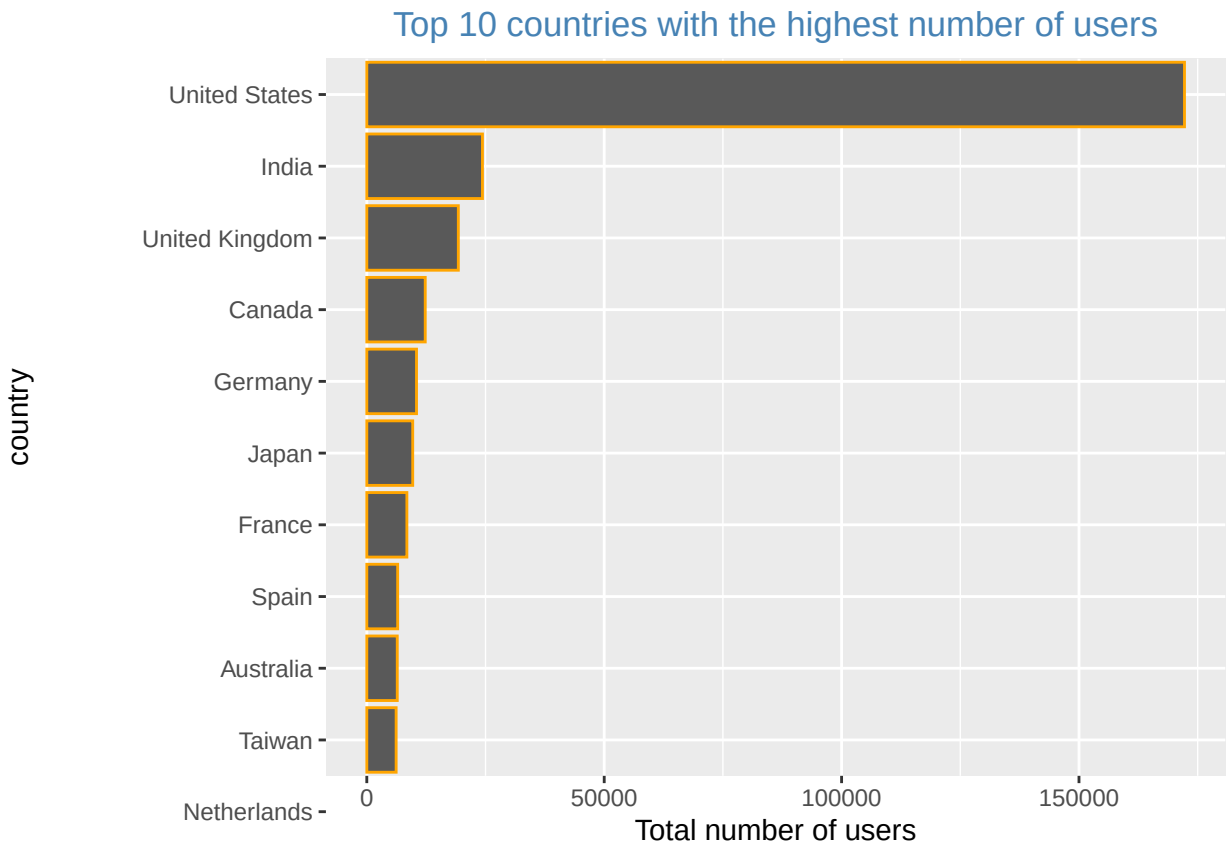
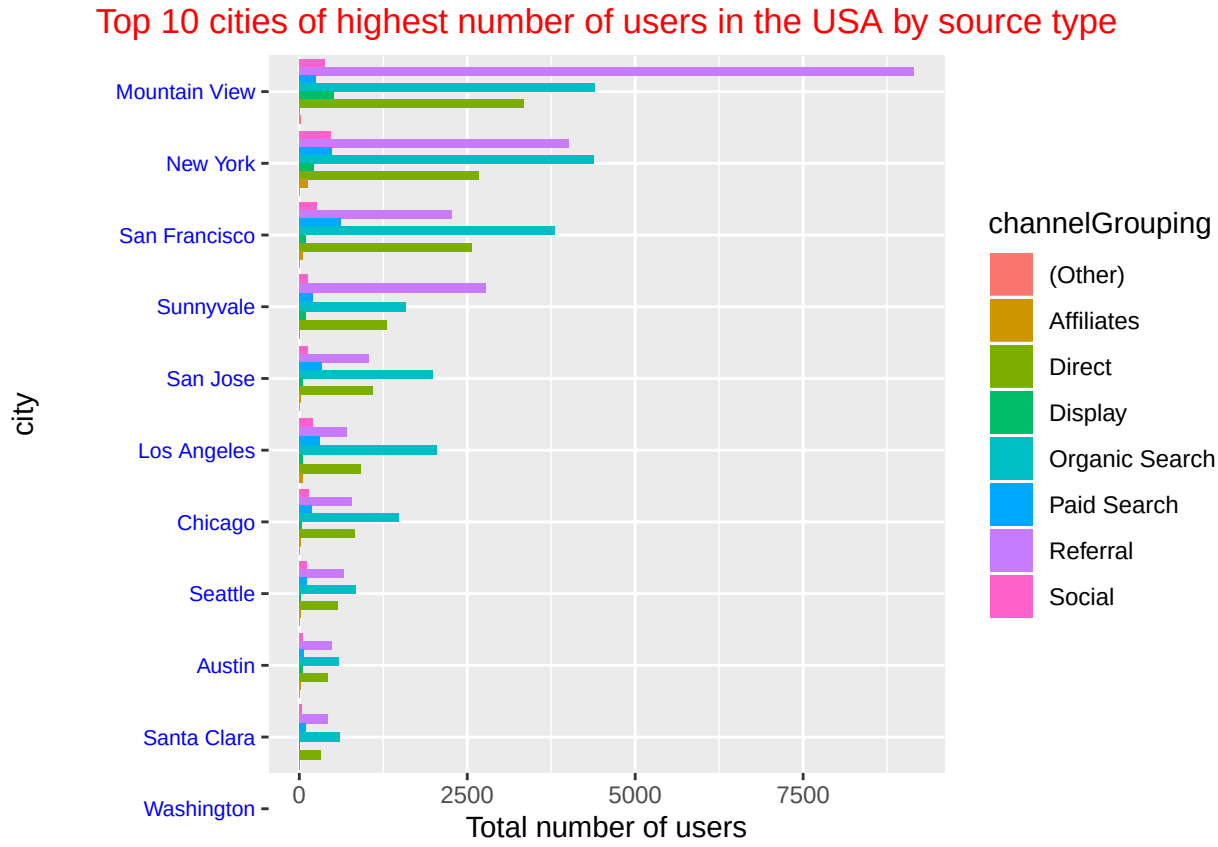
Mightyhive

Draw charts and maps to target areas for marketing campaign

```
data <- read.csv("data.csv") #Read data
data1 <- within(data, country<-factor(country,levels = names(sort(table(country),decreasing = FALSE))))
ggplot(data1)+ aes(x=country)+ geom_bar(position ="dodge",colour="orange")+ ggtitle("Top 10 countries wi
```




```
data_usa <-data1[data1[,"country"] == "United States",]; data_usa <-data_usa[data_usa[,"city"] !="not av
data_usa[is.na(data_usa)]<-0; data_usa1 <- within(data_usa, city<-factor(city,levels = names(sort(table

ggplot(data_usa1)+ aes(city,fill=channelGrouping)+ geom_bar(position ="dodge")+ ggtitle("Top 10 cities
```



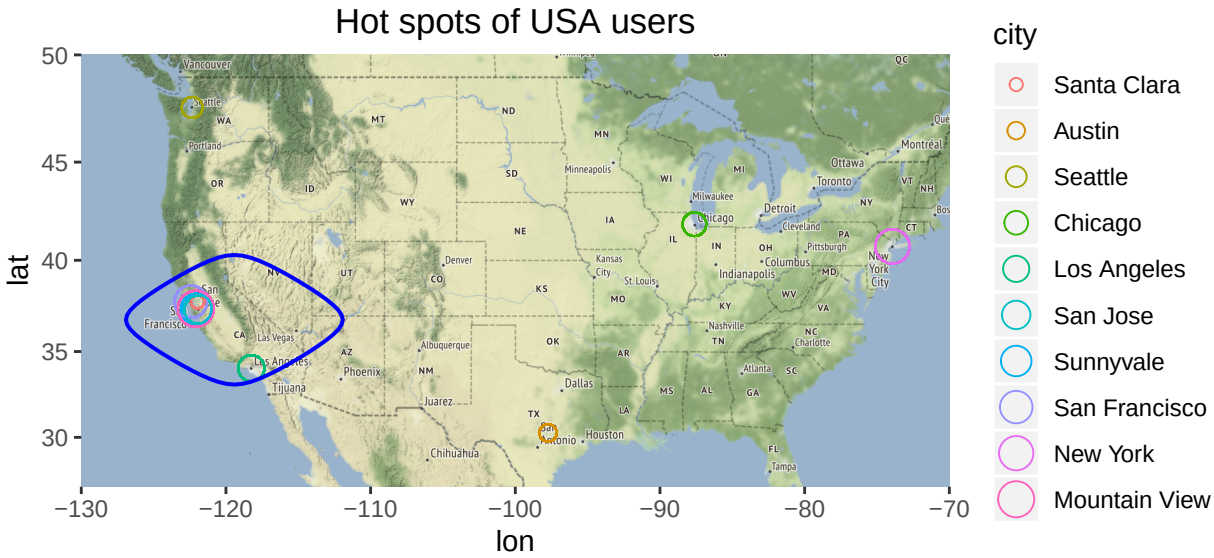

```
data_usa1$lat <-ifelse(data_usa1$city =="Mountain View", 37.386051, ifelse(data_usa1$city == "New York"
data_usa1$long <-ifelse(data_usa1$city =="Mountain View", -122.083855, ifelse(data_usa1$city == "New Yo

ggmap::register_google(key = "AIzaSyDbJlVM2aYCKNe8_YIaMNX5TWx7cZ9kvgs") ##Get Google API
myMap <- get_stamenmap(bbox = c(left = -130, bottom = 27,right = -70, top = 50), maptype = "terrain", c
```

```
## Source : http://tile.stamen.com/terrain/5/4/10.png
## Source : http://tile.stamen.com/terrain/5/5/10.png
## Source : http://tile.stamen.com/terrain/5/6/10.png
## Source : http://tile.stamen.com/terrain/5/7/10.png
## Source : http://tile.stamen.com/terrain/5/8/10.png
## Source : http://tile.stamen.com/terrain/5/9/10.png
## Source : http://tile.stamen.com/terrain/5/4/11.png
## Source : http://tile.stamen.com/terrain/5/5/11.png
## Source : http://tile.stamen.com/terrain/5/6/11.png
## Source : http://tile.stamen.com/terrain/5/7/11.png
## Source : http://tile.stamen.com/terrain/5/8/11.png
## Source : http://tile.stamen.com/terrain/5/9/11.png
## Source : http://tile.stamen.com/terrain/5/4/12.png
## Source : http://tile.stamen.com/terrain/5/5/12.png
```

```
## Source : http://tile.stamen.com/terrain/5/6/12.png
## Source : http://tile.stamen.com/terrain/5/7/12.png
## Source : http://tile.stamen.com/terrain/5/8/12.png
## Source : http://tile.stamen.com/terrain/5/9/12.png
## Source : http://tile.stamen.com/terrain/5/4/13.png
## Source : http://tile.stamen.com/terrain/5/5/13.png
## Source : http://tile.stamen.com/terrain/5/6/13.png
## Source : http://tile.stamen.com/terrain/5/7/13.png
## Source : http://tile.stamen.com/terrain/5/8/13.png
## Source : http://tile.stamen.com/terrain/5/9/13.png
```

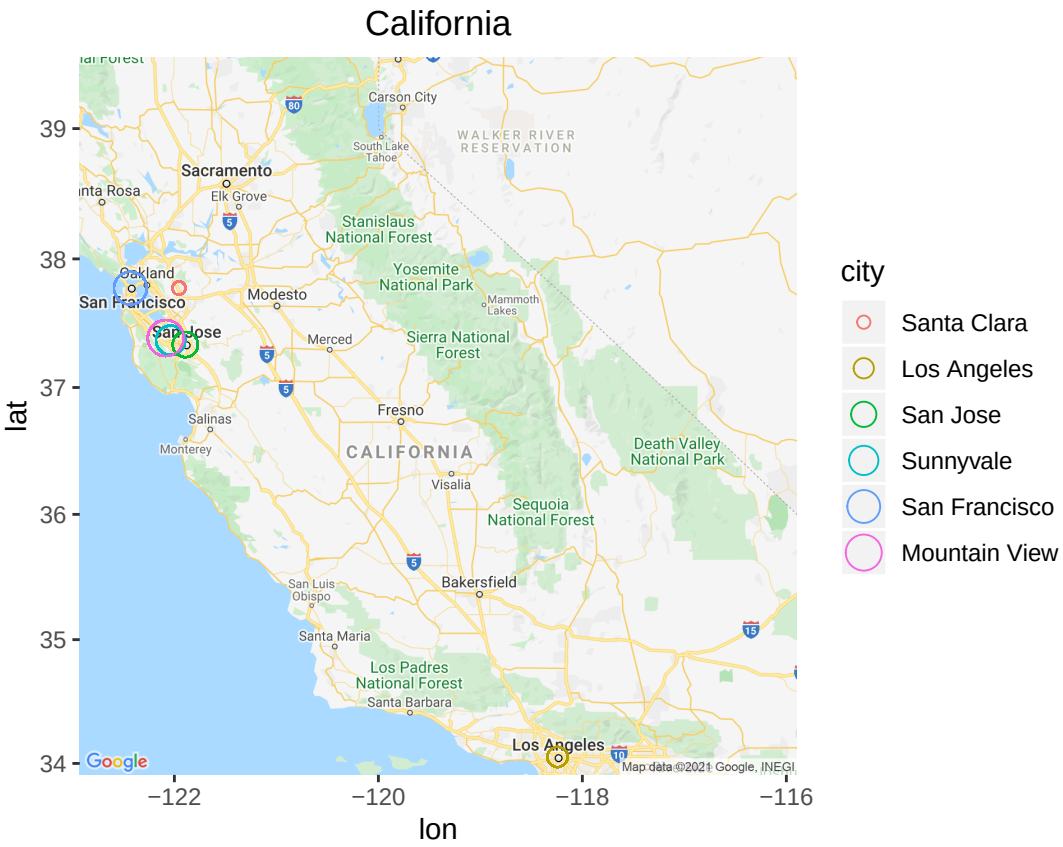
```
ggmap(myMap)+geom_point(aes(x = long, y = lat,size=city,color=city), data = subset(data_usa1,city %in% c
```

```
## Warning: Using size for a discrete variable is not advised.
```




```
ggmap(get_googlemap(center = c(lon =    -119.417931, lat =36.778259),zoom=7,maptype ='roadmap',color =
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=36.778259,-119.417931&zoom=7&size=640
## Warning: Using size for a discrete variable is not advised.
```



Estimate the probability wehter a customer buy in the USA and in California

```
dat <- data_usa; dat$transactions <-ifelse(dat$transactions >=1, 1,0)#Indicate wether a customer buy or
dat_cali <- subset(dat, city %in% c("Mountain View","San Jose","San Francisco","Sunnyvale","Los Angeles
```

Estimate probabilities by general source rules

```
dat_cali_organic <- subset(dat_cali, channelGrouping %in% "Organic Search") #Select only Organic search
convert <- sapply(dat_cali_organic,is.factor)
dat_cali_organic1 <- sapply(dat_cali_organic[,convert],unclass)
dat_cali_organic2 <- cbind(dat_cali_organic[,!convert],dat_cali_organic1) #Change categorical variables

mod = glm(transactions~., data=dat_cali_organic2)
best_model <- step(mod,direction ="both")#get the best model
```

```
## Start:  AIC=-22450.75
## transactions ~ fullVisitorID + visitNumber + date + bounces +
##     hits + pageviews + timeOnSite + transactionRevenue + source +
##     channelGrouping + browser + deviceCategory + country + city
##
##
## Step:  AIC=-22450.75
## transactions ~ fullVisitorID + visitNumber + date + bounces +
##     hits + pageviews + timeOnSite + transactionRevenue + source +
##     channelGrouping + browser + deviceCategory + city
##
##
## Step:  AIC=-22450.75
```

```
## transactions ~ fullVisitorID + visitNumber + date + bounces +
##     hits + pageviews + timeOnSite + transactionRevenue + source +
##     browser + deviceCategory + city
##
##                       Df Deviance    AIC
## - browser              1   177.93 -22453
## - fullVisitorID        1   177.94 -22453
## - date                 1   177.95 -22451
## <none>                     177.93 -22451
## - visitNumber          1   177.99 -22449
## - timeOnSite           1   178.02 -22446
## - city                 1   178.03 -22445
## - source               1   178.04 -22444
## - deviceCategory       1   178.18 -22433
## - bounces              1   179.26 -22345
## - hits                 1   180.39 -22255
## - pageviews            1   183.81 -21984
## - transactionRevenue   1   197.28 -20963
##
## Step:  AIC=-22452.71
## transactions ~ fullVisitorID + visitNumber + date + bounces +
##     hits + pageviews + timeOnSite + transactionRevenue + source +
##     deviceCategory + city
##
##                       Df Deviance    AIC
## - fullVisitorID        1   177.94 -22455
## - date                 1   177.95 -22453
## <none>                     177.93 -22453
## + browser              1   177.93 -22451
## - visitNumber          1   177.99 -22451
## - timeOnSite           1   178.02 -22448
## - city                 1   178.03 -22447
## - source               1   178.04 -22446
## - deviceCategory       1   178.28 -22427
## - bounces              1   179.26 -22347
## - hits                 1   180.39 -22257
## - pageviews            1   183.81 -21986
## - transactionRevenue   1   197.28 -20965
##
## Step:  AIC=-22454.47
## transactions ~ visitNumber + date + bounces + hits + pageviews +
##     timeOnSite + transactionRevenue + source + deviceCategory +
##     city
##
##                       Df Deviance    AIC
## - date                 1   177.96 -22455
## <none>                     177.94 -22455
## + fullVisitorID        1   177.93 -22453
## + browser              1   177.94 -22453
## - visitNumber          1   177.99 -22452
## - timeOnSite           1   178.03 -22449
## - city                 1   178.03 -22449
## - source               1   178.04 -22448
## - deviceCategory       1   178.28 -22429
```

```
## - bounces             1   179.27 -22349
## - hits                1   180.39 -22259
## - pageviews           1   183.82 -21987
## - transactionRevenue  1   197.29 -20967
##
## Step:  AIC=-22454.9
## transactions ~ visitNumber + bounces + hits + pageviews + timeOnSite +
##     transactionRevenue + source + deviceCategory + city
##
##                       Df Deviance    AIC
## <none>                     177.96 -22455
## + date                 1   177.94 -22455
## + fullVisitorID        1   177.95 -22453
## + browser              1   177.96 -22453
## - visitNumber          1   178.01 -22453
## - timeOnSite           1   178.05 -22450
## - city                 1   178.05 -22450
## - source               1   178.09 -22446
## - deviceCategory       1   178.30 -22429
## - bounces              1   179.30 -22348
## - hits                 1   180.41 -22260
## - pageviews            1   183.82 -21989
## - transactionRevenue   1   197.31 -20967
```

```
dat_cali_organic3<- dat_cali_organic %>% dplyr::select(transactions,date, bounces, hits, pageviews, time
inTrain_cali_organic = createDataPartition(y = dat_cali_organic$transactions,p = 0.75, list = FALSE) #G

TrainingSet_cali_organic= dat_cali_organic3[inTrain_cali_organic, ]
TestSet_cali_organic= dat_cali_organic3[-inTrain_cali_organic, ]

model.cali.organic = glm(transactions~., data=dat_cali_organic3)

ptest_cali_organic  = predict(model.cali.organic, newdata = TestSet_cali_organic); ptrain_cali_organic =

o_search <- mean(ptrain_cali_organic) #organic search probability, mean(ptest_cali_organic): test sets


###From this lines there will be exactly same formats for coding.Only source type would be different. So

dat_cali_Referral <- subset(dat_cali, channelGrouping %in% "Referral"); convert <- sapply(dat_cali_Refe

dat_cali_Direct <- subset(dat_cali, channelGrouping %in% "Direct");convert <- sapply(dat_cali_Direct,is
model.cali.Direct = glm(transactions~., data=dat_cali_Direct3);ptest_cali_Direct  = predict(model.cali.
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
dat_cali_Paid <- subset(dat_cali, channelGrouping %in% "Paid Search");convert <- sapply(dat_cali_Paid,is
model.cali.Paid = glm(transactions~., data=dat_cali_Paid3);ptest_cali_Paid  = predict(model.cali.Paid, n

dat_cali_Social <- subset(dat_cali, channelGrouping %in% "Social");convert <- sapply(dat_cali_Social,is
```

```
dat_cali_Display <- subset(dat_cali, channelGrouping %in% "Display");convert <- sapply(dat_cali_Display

df <- data.frame(Source = rep(c("Organic search", "Referral", "Direct" ,"Paid Search","Social","Display

ggplot(df, aes(x = reorder(Source,-Probability), y = Probability))+ geom_bar(stat ="identity")+labs(x="
```