osXtern-Analysis

Daniel Hauersperger

10/23/2017

I've been given a dataset and a mission to come up with ideas for new features for the next edition of osXtern, the operating system for Xterns. Let's jump right in and load the csv.

```
checkinData <- read.csv("/Users/danielhauersperger/Downloads/checkin_dataset.csv")
```

Now let's look at what data we'll be working with

```
checkinData[1:10,]
```

```
##    X user           timestamp xcoordinate ycoordinate
## 1  0   12 2017-07-13 09:36:00   0.9068354   0.7764837
## 2  1   12 2017-07-30 15:23:00   0.9285871   0.8049635
## 3  2   12 2017-05-05 00:41:00   0.9040911   0.7840430
## 4  3   12 2017-07-26 06:10:00   0.9067523   0.8044615
## 5  4   12 2017-05-22 13:22:00   0.9006413   0.7816826
## 6  5   12 2017-06-11 13:05:00   0.8996803   0.7908925
## 7  6    8 2017-07-13 09:40:00   0.8737654   0.8087066
## 8  7    8 2017-07-30 15:23:00   0.9134757   0.7897425
## 9  8    8 2017-05-05 00:43:00   0.9152557   0.7906853
## 10 9    8 2017-07-26 06:12:00   0.9162729   0.7858601
```

It looks like we have a unique identifier called X, a number corresponding to a user, a timestamp, and x and y coordinates.

```
nrow(checkinData)
```

```
## [1] 25668
```

It appears each row corresponds to one observation of a user's location at a given time, so we have 25,668 observations.

Let's make sure X really is a unique identifier.

```
length(checkinData$X) == length(unique(checkinData$X))
```

```
## [1] TRUE
```

The number of X values is the same as the number of unique X values, so there are no repeats and X is unique for each observation. Now I'm curious about how many users there are.

```
length(unique(checkinData$user))
```

```
## [1] 100
```

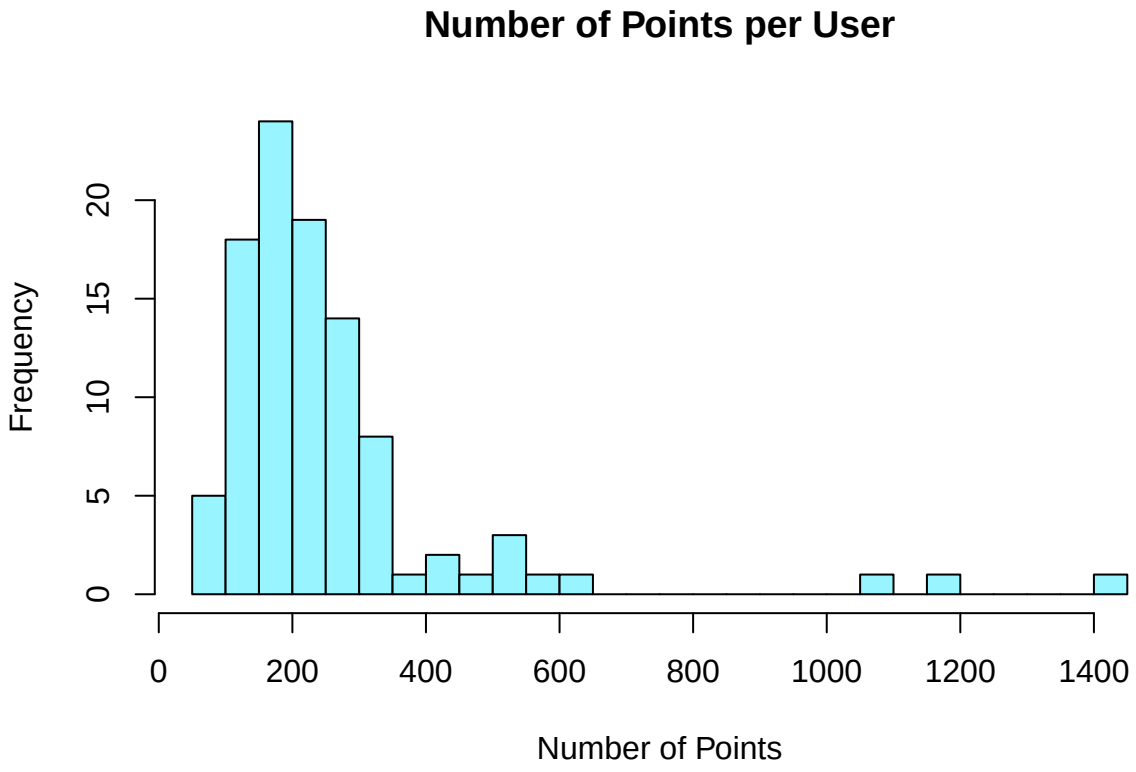
Hence there are 100 users. Now I'd like to see how many observations we have for each user.

```
library(dplyr)
```

```
userPoints <- count(checkinData, user) # Count how many data points we have for each user
range(userPoints$n)
```

```
## [1]   77 1426
```

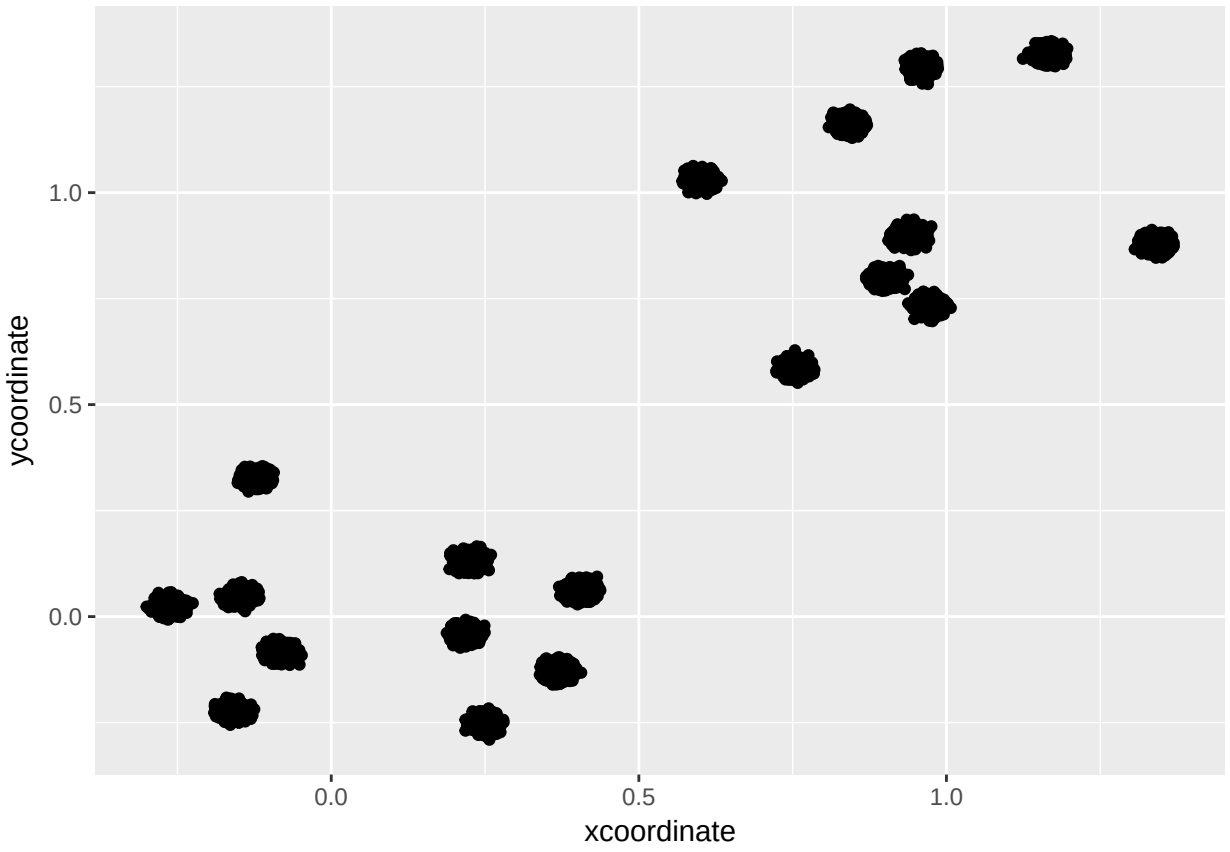
```
hist(userPoints$n, breaks = 30, main = "Number of Points per User", xlab = "Number of Points", col = "ca
```



So we have between 77 and 1426 data points for each user, with most users having around 200 data points according to the above histogram. Let's go ahead and plot the locations to see where the users are checking in.

```
library(ggplot2)
```

```
ggplot(aes(x = xcoordinate, y = ycoordinate), data = checkinData) + geom_point()
```

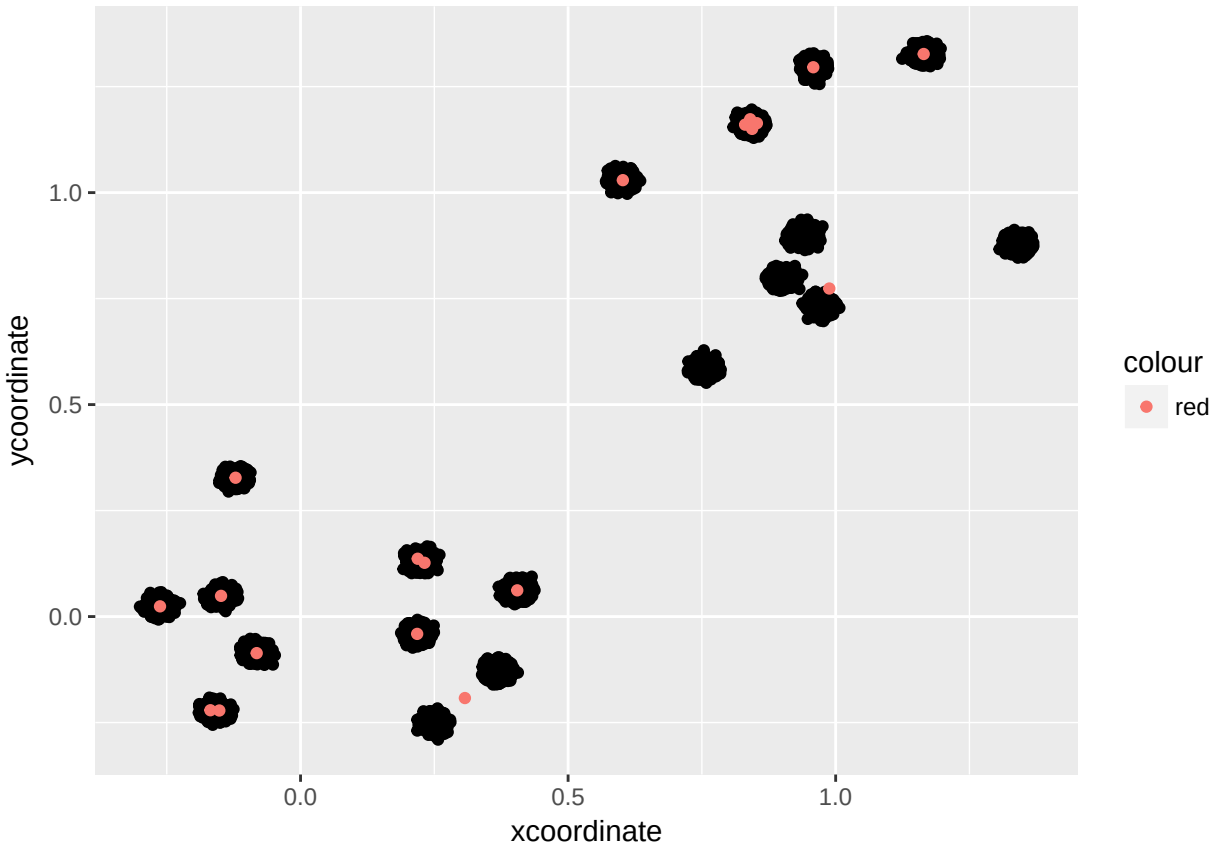


It looks like there are 19 locations (clusters) where the users tend to check in to the app. Perhaps there are 19 specific buildings or areas where Xterns tend to spend time. I'd like to group their locations based on these clusters, so I'll first try k-means clustering.

```
kmeansResults <- kmeans(cbind(checkinData$xcoordinate, checkinData$ycoordinate), 19)
kmeansCenters <- as.data.frame(kmeansResults$centers) # store the center coordinates as a dataframe nam
names(kmeansCenters) # Check the column names so I can use them in the next line
```

```
## [1] "V1" "V2"
```

```
ggplot(aes(x = xcoordinate, y = ycoordinate), data = checkinData) + geom_point() + geom_point(aes(x = V
```



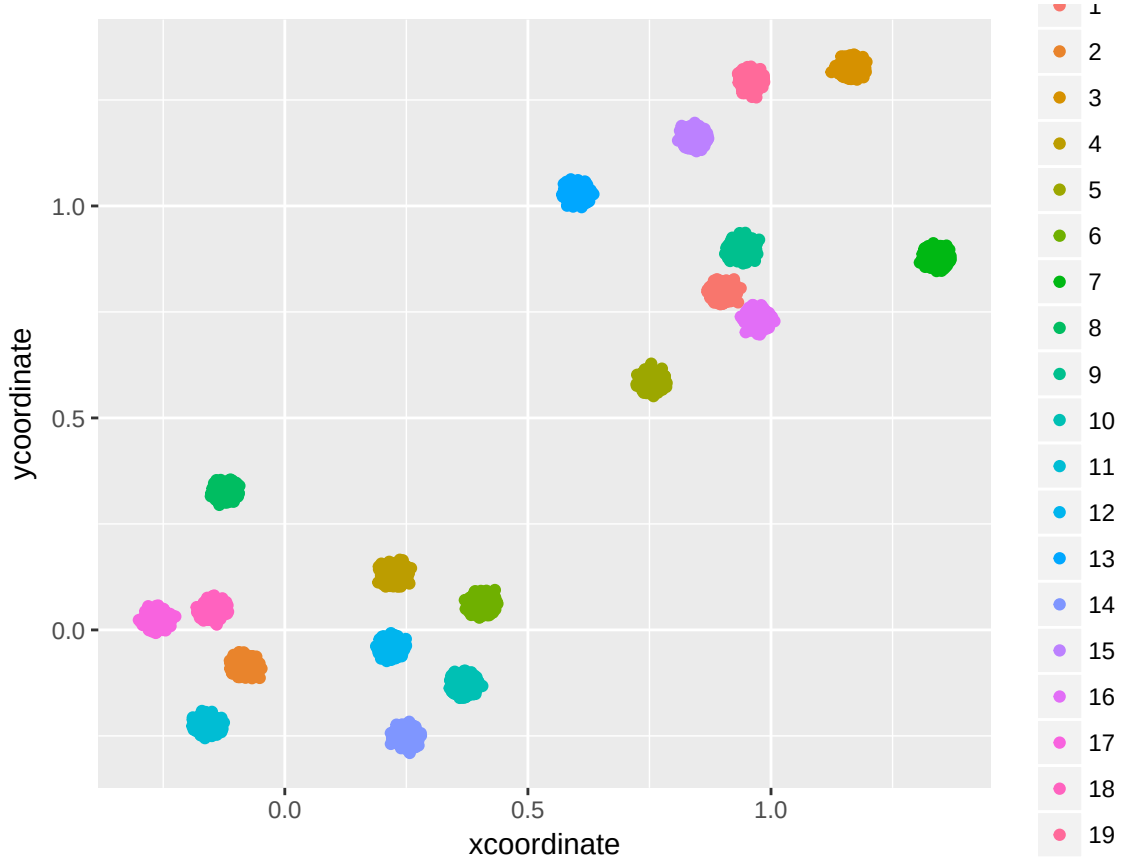
Based on the above plot it appears several points were not classified correctly (I was hoping to see one center in each of the black splotches), but the model is likely stuck in a local minimum. I tried several different approaches (not pictured), such as randomly generating several sets of initial centers, but none were fruitful. I then found and tried a density-based approach called dbscan.

```
library(dbscan)
dbscanResults <- dbscan(cbind(checkinData$xcoordinate, checkinData$ycoordinate), eps = 0.02, minPts = 19
# I tried a few other values for eps, but settled on eps = 0.02 as it yielded the desired 19 clusters
dbscanResults
```

```
## DBSCAN clustering for 25668 objects.
## Parameters: eps = 0.02, minPts = 19
## The clustering contains 19 cluster(s) and 0 noise points.
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 1313 1367  914 2205 1461 1558 1517  971 1219 1219  988 1603 1115 1334 1491
##   16   17   18   19
## 1540 1333 1154 1366
##
## Available fields: cluster, eps, minPts
```

As you can see there are no noise points, so there are no outliers far enough from a cluster to be removed. This makes me feel better about the approach. Now let's look at a graph.

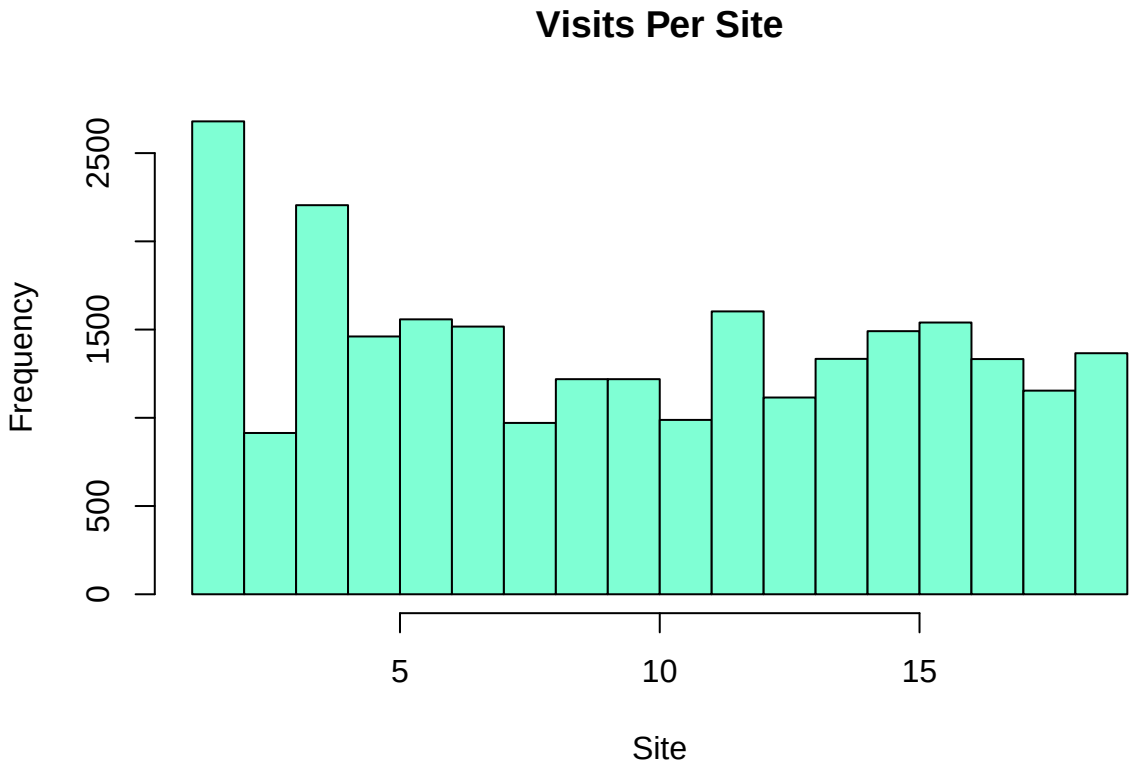
```
checkinData$site <- dbscanResults$cluster
ggplot(aes(x = xcoordinate, y = ycoordinate, color = as.factor(site)), data = checkinData) + geom_point
```



As you can see, each of the 19 clusters is a unique color; combining this with the knowledge that there are no outliers, we can see that all observations are properly assigned to a cluster. I already added the groups to the data frame, so let's go on and see what else we can find out.

First off let's see if any of the sites had a particulary high number of visits.

```
hist(checkinData$site, main = "Visits Per Site", xlab = "Site", col = "aquamarine")
```



Based on the above histogram it looks like all 19 sites were visited many times, with sites 1 and 3 leading the pack in number of visits. Because sites 1 and 3 see many visitors, we may consider recommending them early on to new Xterns next summer so they can become familiar with the most popular places early.

Speaking of early, let's start looking at the timestamp data.

```
timestamps <- as.POSIXlt(checkinData$timestamp) # view the timestamps as actual r timestamps
min(timestamps)
```

```
## [1] "2017-05-01 00:26:00 EDT"
```

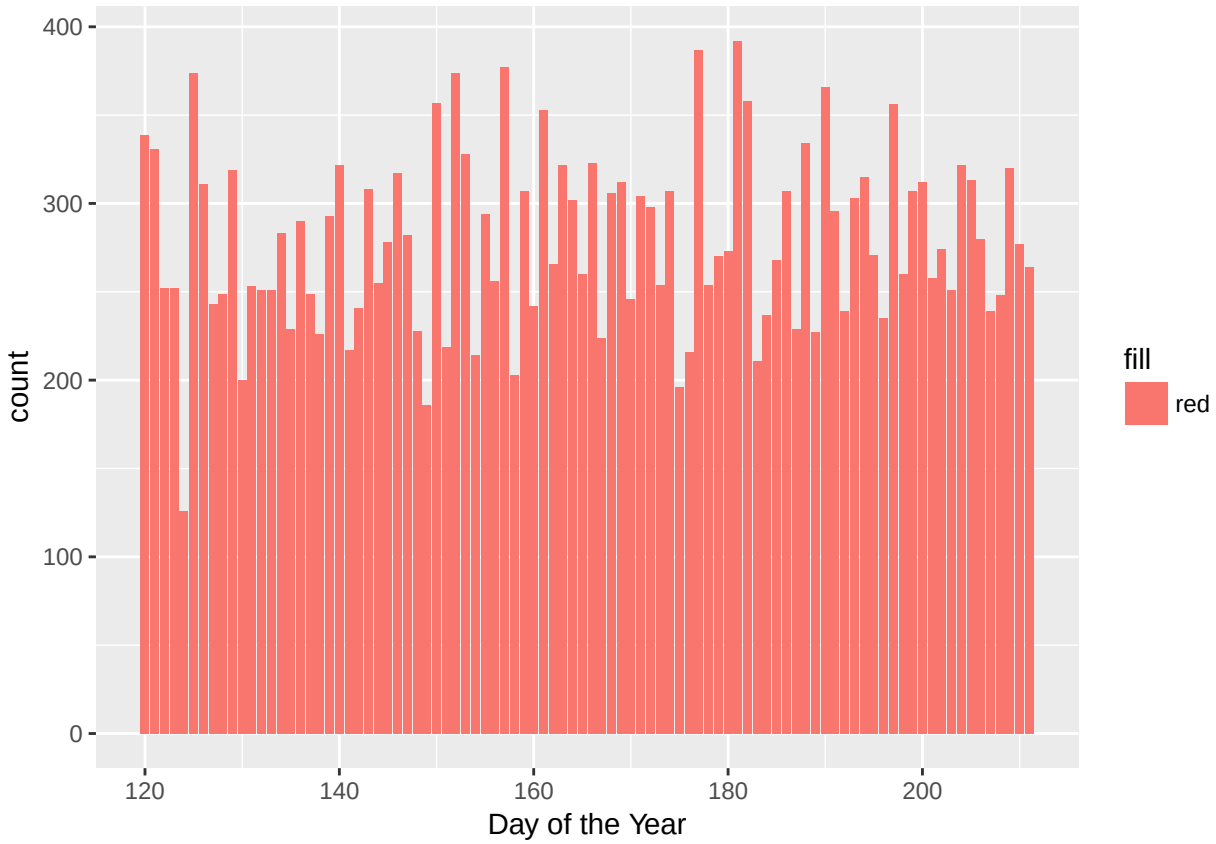
```
max(timestamps)
```

```
## [1] "2017-07-31 23:28:00 EDT"
```

The timestamps go from May 1 to July 31 2017, which makes sense as that is the time covered by the Xtern program last year.

I'm interested if we can see some trends in when the users give their location. Let's first see if one day is particularly popular.

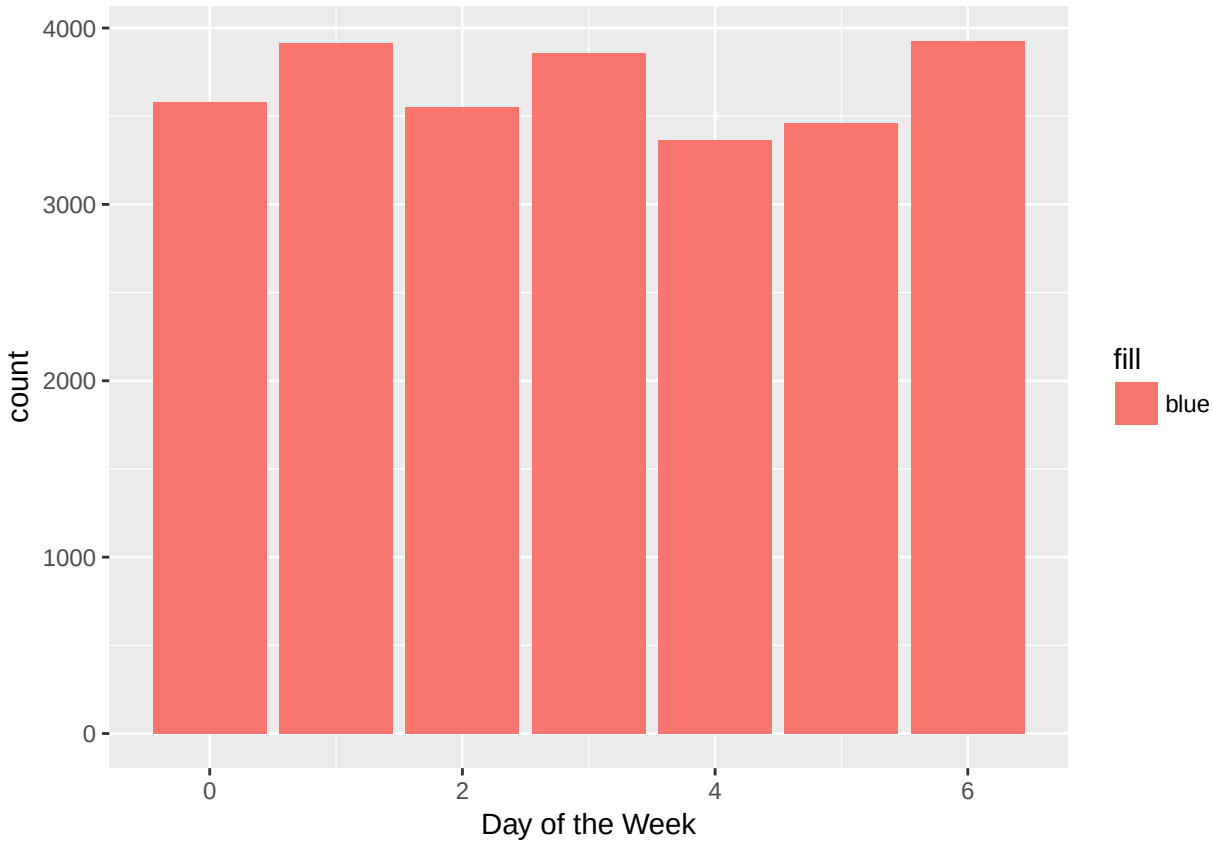
```
checkinData$dayOfYear <- as.POSIXlt(checkinData$timestamp)$yday ## add a column with the day of the yea
ggplot(aes(x = as.POSIXlt(timestamp)$yday, fill = "red"), data = checkinData) + geom_bar() + labs(x = "
```



Based on the histogram, it looks like the visits for each day of the summer are approximately equal. This is somewhat promising as it show that the Xterns became engaged in osXtern quite early in the summer, and their activity stayed reasonably consitent over time. It's nice to see that there isn't a downward trend in user engagement over time.

Let's check if any particular day of the week is more popular than the others.

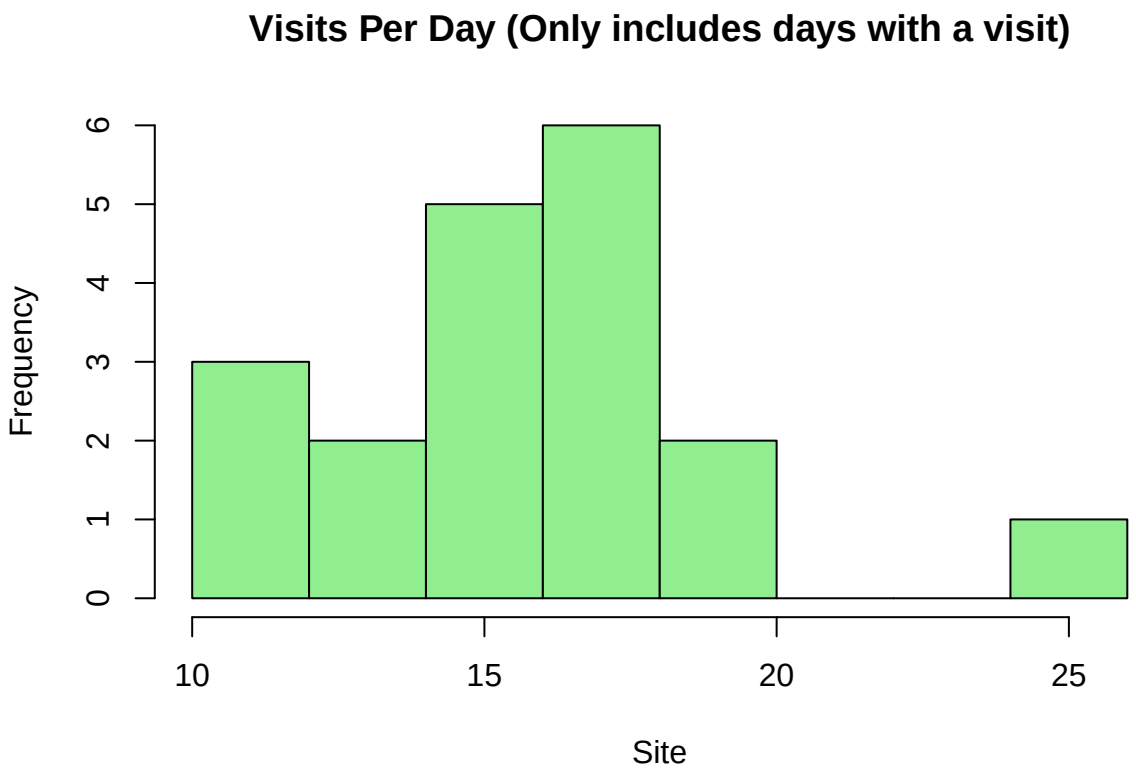
```
ggplot(aes(x = as.POSIXlt(timestamp)$yday %% 7, fill = "blue"), data = checkinData) + geom_bar() + labs
```



It looks like the number of visits on each day of the week is approximately equal as well. It's good to see the Xterns were engaged consistently throughout the week, but I want to dig a bit deeper.

I'll next check how many people visited each site at a time.

```
## find average number of people to visit each place on a day in which there are visits
numPeopleTemp <- checkinData %>% group_by(site, dayOfYear) %>% summarise(n = n()) # get tibble (special
numPeople <- numPeopleTemp %>% group_by(site) %>% summarise (mean1 = mean(n)) # take the mean of the co
hist(numPeople$mean1, main = "Visits Per Day (Only includes days with a visit)", xlab = "Site", col = "
```



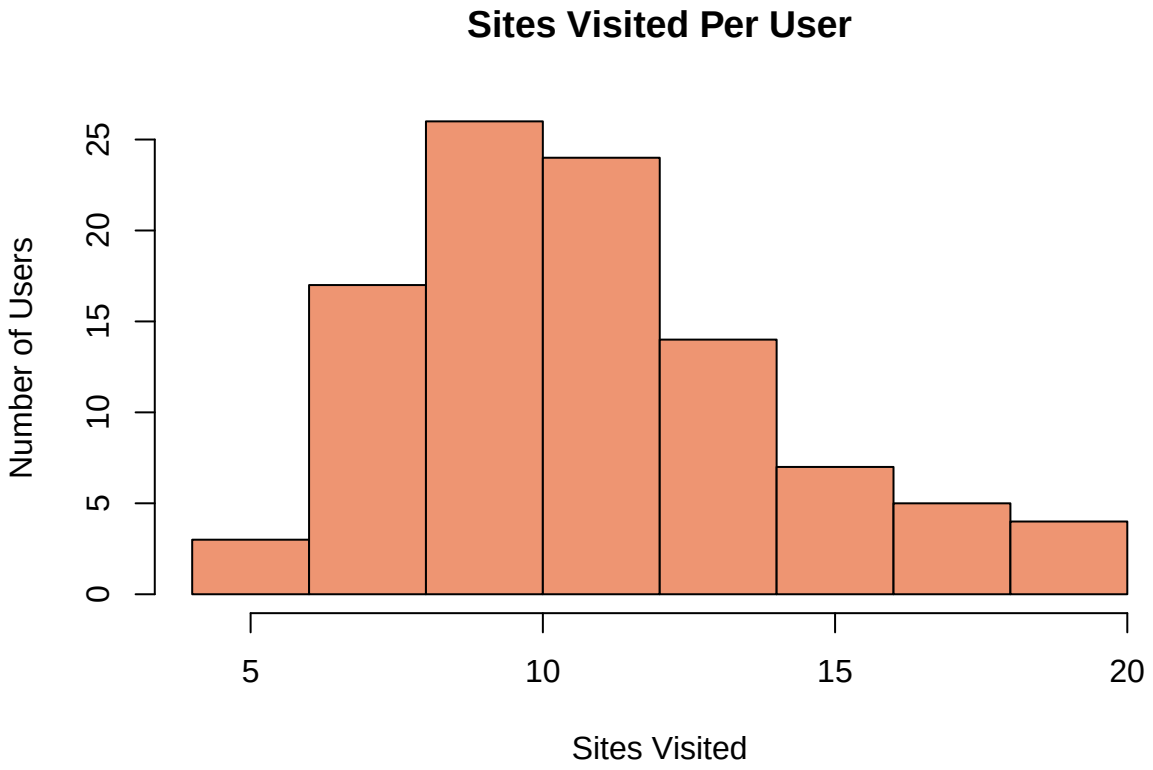
Based on this histogram, we can see there is one location that sees relatively large groups of people (on the days people visit). Let's see what group this is.

```
numPeople[numPeople$mean1 == max(numPeople$mean1),]$site
```

```
## [1] 4
```

We can now see this location is number 4. Because site 4 tends to see large numbers of people, perhaps osXtern could suggest that people who aren't as involved should visit site 4 when it sees a group forming in order to boost morale.

```
## Check how many sites each user visits throughout the summer
numSitesPerUser <- checkinData %>% group_by(user) %>% summarise(n = n_distinct(site))
hist(numSitesPerUser$n, col = "lightsalmon2", main = "Sites Visited Per User", xlab = "Sites Visited", y
```



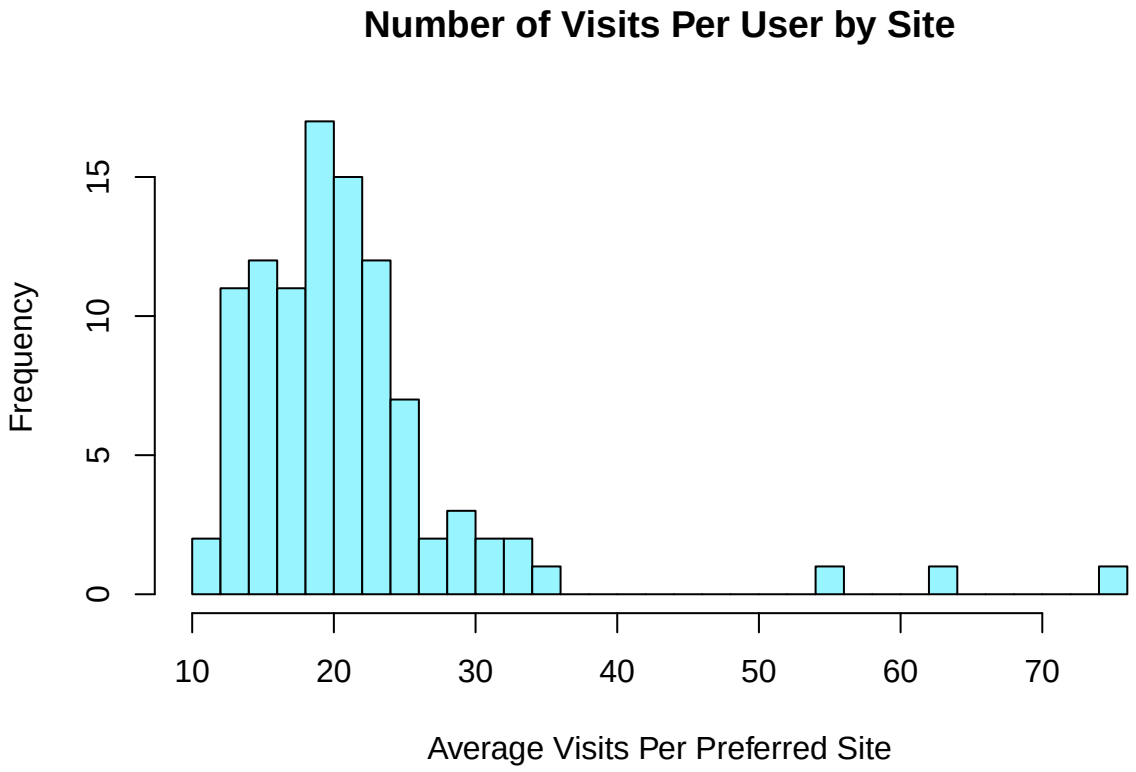

```
range(numSitesPerUser$n)
```

```
## [1]  5 19
```

We can see that most users tend to visit around 10 sites during the course of the summer, but some users visit as few as 5 sites or as many as 19 sites. Perhaps we could encourage users that visit less sites to try out a couple of new ones. We could even connect them with users that visit a lot of sites.

I next wanted to see how many times users visited each site.

```
numVisitsEachSite <- checkinData %>% group_by(site, user) %>% summarise(n = n()) # get a tibble with ho
numVisitsEachSiteMean <- numVisitsEachSite %>% group_by(user) %>% summarise (mean1 = mean(n)) # take th
hist(numVisitsEachSiteMean$mean1, breaks = 30, col = "cadetblue1", main = "Number of Visits Per User by
```



Based on the histogram we can see that the average person visited each of their preferred sites between roughly 15 and 30 times, but a few people visited their preferred sites more than 50 times during the summer (I use preferred in this case to show the sites that the user actually visited). From this we can see that Xterns tend to be quite loyal to their favorite sites, so when we try to help user engagement by advertising nearby sites and events, we should be sure to do it in a way that doesn't insult the sites they are currently visiting.

Final Thoughts

We were given a sample dataset and with only our minds, computers, and the wealth of all human knowledge at our fingertips, we were able to make several interesting findings. I'll now describe how our findings can be used to shape decisions in the future.

We observed that all users in the sample used their location at least 77 times, with most using it around 200 times. This shows that our users are active at least once per day in general, so we may not have to be incredibly worried about their engagement frequency, and instead should focus on the value added by each engagement.

We found that all of the sites at which users gave their location could be grouped neatly into 19 clusters. There are many more than 19 interesting places to visit in Indy, so I suggest that we look for a few more places to visit. We can then recommend these places to the Xterns throughout the summer to facilitate them exploring this great city.

We discovered that most of the sites were roughly equally popular, but sites 1 and 3 seem to be especially beloved. I recommend that next summer we advertise sites 1 and 3 to start the summer off with a bang.

We showed no discernible trend in user engagement over time. With more time to invest in studying the data we may be able to find some trends, but for now it is nice to know that our users stayed active throughout the summer. This should shape osXtern's future development by reminding us to at least keep the current features, as they don't seem to lose their appeal in time.

We found on days that sites are visited, they tend to be visited around 15 times. This leads me to think that

Xterns travel in packs, so we should design features to complement their interactions, such as automatically giving a group discount when enough Xterns visit a restaurant at once. On this note we also showed that site 4 is visited by quite large groups of Xterns. Perhaps site 4 is a large venue, so we should suggest it to Xterns who are gathered in a medium-sized group elsewhere.

We next showed that Xterns tended to visit around 10 sites throughout the summer. This is good, but with some users visiting as low as 5 sites, I think we can do better. Perhaps next summer if we observe users are trying only a few sites, we can offer incentives such as coupons to encourage them to try other places.

Finally, we showed how many times each user visited each of their preferred sites. I found that Xterns were loyal to their favorite sites, so we should be careful not to insult their loyalty when suggesting new sites. Perhaps we should be careful not to recommend new sites when they are currently at one of their preferred sites, and instead wait until times they have not checked in for a while.

I'm sure I could have come up with more insights with more time to explore the data. A few of the questions I would have liked to ask are the following:

1. What is the average group size for each user? Do people who stay in small groups tend to hang out together?
2. Do some users visit some sites more often than others?
3. Is there a connection between number of sites visited and number of visits per site for a given user?
4. How many locations does an average user visit?
5. Was I justified in using the days of the visits, and not specifically using the minutes/seconds?
6. Do some users tend to stay further away from the center of a location?

Despite having these unanswered questions remaining, I'm quite happy with the progress I made.

Thanks for letting me lead you along this journey through the data. Have a good day, and bye for now.