KPI analysis for a mobile application

Background

This report is created for a product owner who is charge of running the development team for a mobile application. The aim of the analysis is to a better understanding of user behaviours when interacting with his product. The app is in its 3rd version currently and an analysis was carried out to compare the difference in some key KPI statistics between older version and a current version.

Technologies used

This analysis was carried out using Pandas in Python. Matplotlib style visualisations were used to generate graphical summaries. Although it is understood that some queries on slicing of data would be performed quicker and easier in SQL, Python was used throughout the investigation to keep the code tidy and consistent.

Assumptions and checks

It is generally assumed that the collected data are free from errors. It is also assumed that the data is available is all (or at least the majority) of the information available and reflects the real situation of the app usage.

Loading and initial inspection of the users data

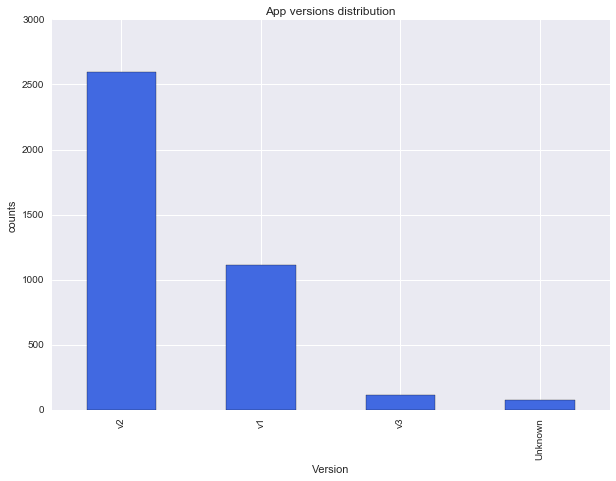
The data was separated into 2 CSV files namely “users” and “events”. Both files were loaded and analysed. There are 3900 rows in the “users” file and 3 columns. The columns recorded the user ID, app installation date and version installed respectively. A quick check shows that there are no duplicates in user ID as expected. The earliest installation date recorded was 1st December 2014 and the latest installation date was 31st July 2016. Fig.1 shows the distribution of versions installed.

Fig.1 Number of installation for each version of the app

It can be seen that there are unknown versions in the data. It should be possible to estimate app versions for those unknown instances using install date.

After performing a few enquires, it can be seen that there is a total of 75 unknowns. We could divide the data into 3 time periods. Period 1 denotes the period when only v1 was available. Period 2 starts when the first v2 was being downloaded and ends when the first v3 was downloaded. Period 3 consists of everything after the first v3 was downloaded. It can be safely assumed that the unknowns in Period 1 use v1 of the app. However the tricky situation arises when v2 was made available, v1 was not discontinued. According to the ratio in period 2, there should be 2-3 v1 users and over 50 v2 users in the unknown group. However it would be nearly pointless to estimate the unknown in this way. In order to solve this problem we can either assume them to be all v2 users or we can try to do a better estimation when the 2 data sets are combined. There are 5 unknowns in period 3 and we face a similar problem as we did for period 2.

For preliminary analysis, it would be acceptable to drop the unknown cases for simplicity. This is because we have quite a large group of v1 and v2 users (where the majority of the unknown belongs). v3 users number can be a concern but estimating the unknowns would not help in this scenario.

There are 27726 rows and 5 columns in the events data. The columns represent the following information:

* User ID
* Session ID
* Page visited
* Time spent in page
* Date and time of leaving the page

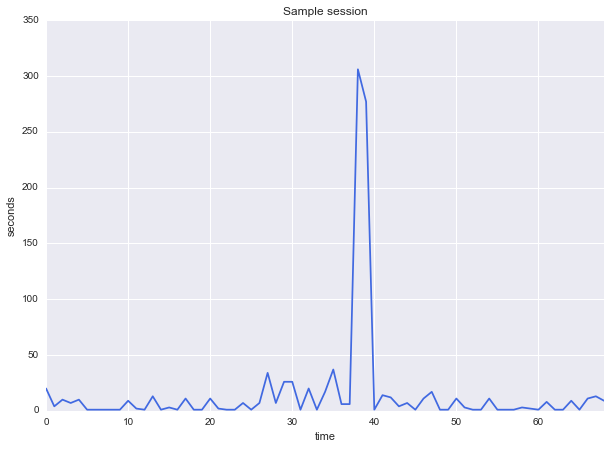


Fig.2 Events duration within a session

The number of unique users does not match up with the number in the user data set. Therefore it can be assumed that there are cases where customers downloaded the app but have not actually used it.

It would also be helpful to clarify the difference between session and page visits (events). There could be multiple events in a unique session. As long as users stay in the app it is considered a single session. However the user can visit different parts of the app and when they leave that part an event would be triggered. Fig.2 in the previous page demonstrates a series of events in a single session. It can be seen that some of the session durations are very short. It doesn't look like a normal usage. It looks more like a bug or users forgetting to lock their screen.

Similar to the installation date, the date and time for the first and last recorded events can be founded. The first recorded event was in March 2015, about 3 months after the first installation. The last recorded event was in October 2016.

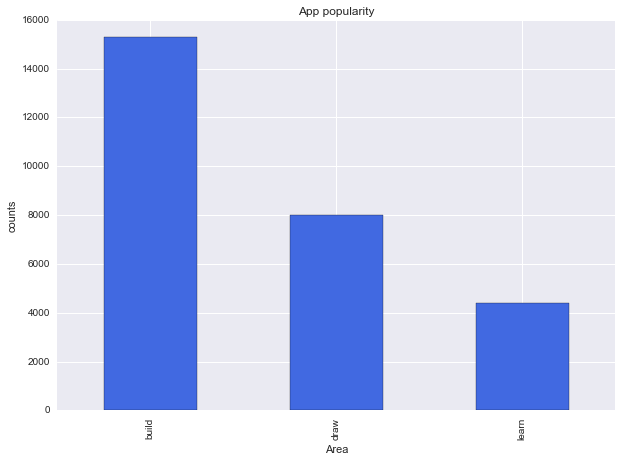


Fig.3 Number of events for different pages

Fig. 3 shows the popularity of the individual page/section of the app. It seems that 'build' is the most popular area of the app. However we should take into account the phenomenon discovered in the previous part. Simply counting the number of events might not tell the true story when bugs or unintended operations exist.

Fig.4 on the next page shows the distribution of session length, It can be seen that time spent in each event can be as short as 1 second. These short sessions do not seem like normal usage. Max time spent is 7195 seconds, which equals to about 120 minutes. I would say it is a reasonable time to spend in an app if you are really engaged. Therefore to conclude, the time data is reasonable and can be trusted. However that doesn't necessarily mean they reflect normal user operations.

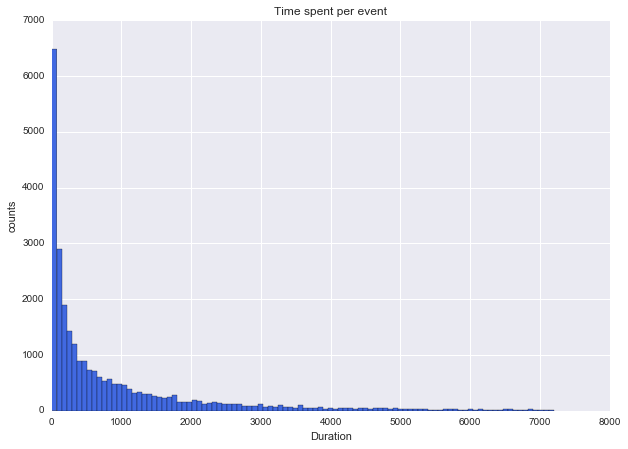


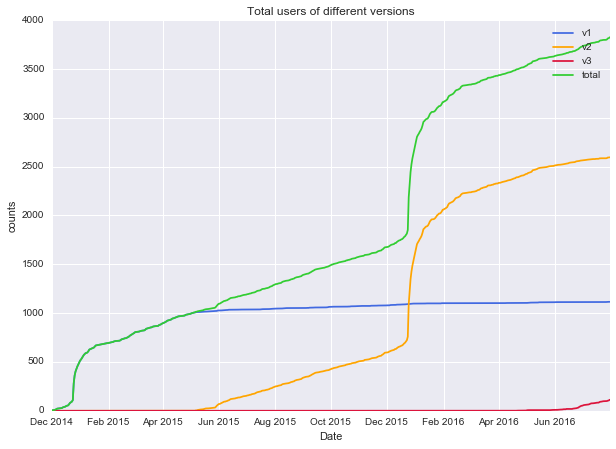
Fig.4 Duration of sessions

To conclude, this session we have loaded and inspected the data. Apart from the entries that have unknown app versions, all the data looks reasonable. Some preliminary exploration was also carried out to get a feel of what the data is like. In the following sessions, it would be logical to look at some common KPIs to gain a deeper understanding of the app.

Total Users

Total user number is cumulative measure across a period of time. Therefore it is expected to always see an upward sloping line no matter what the gradient is. This metric is useful to see how well the app/company is doing in terms of gathering new users and how the user base changes in size. This in turns affects the firm's marketing strategy and other important decisions such as server capacities.

Fig.5 on the next page shows a figure that demonstrates the changes in total users for the different versions of the app. From the plot it can be seen that the app had 2 rapid growing phases (late Dec 2014 and late Dec 2015). The growth in between was very steady. We can see that v2 of the app was launched in May 2015 and takes over from v1. However there is no significant change in the app installation rate. There is another large increase in installation between late Dec 2015 and Jan 2016. It could be interesting and worthwhile to further investigate the cause of the change.



Monthly Active Users

Fig.5 Change in total users

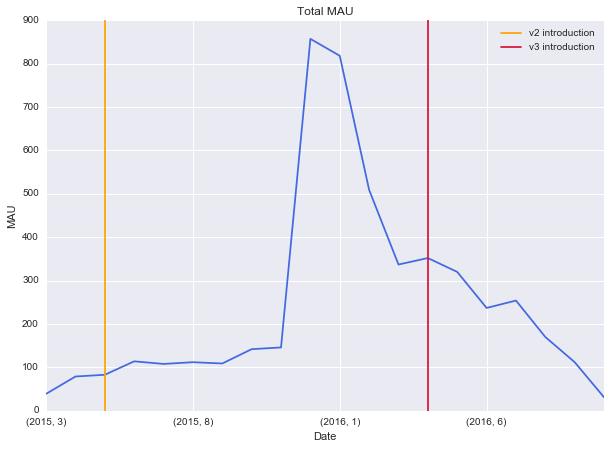
This metric measures the number of unique users that have used the app within a month. This is an important metric as it tells us how popular the app is. It should be noted that in our investigation, 2015 March and 2016 Oct are not complete months. The results were plotted and shown in Fig.6

Fig.6 Application MAU

Merging the datasets allow us to look at MAU for individual versions of the app as seen in Fig.7. User is merged into event using a left join. This is because we are mainly interested in the number of unique users that have used the app but not all users.

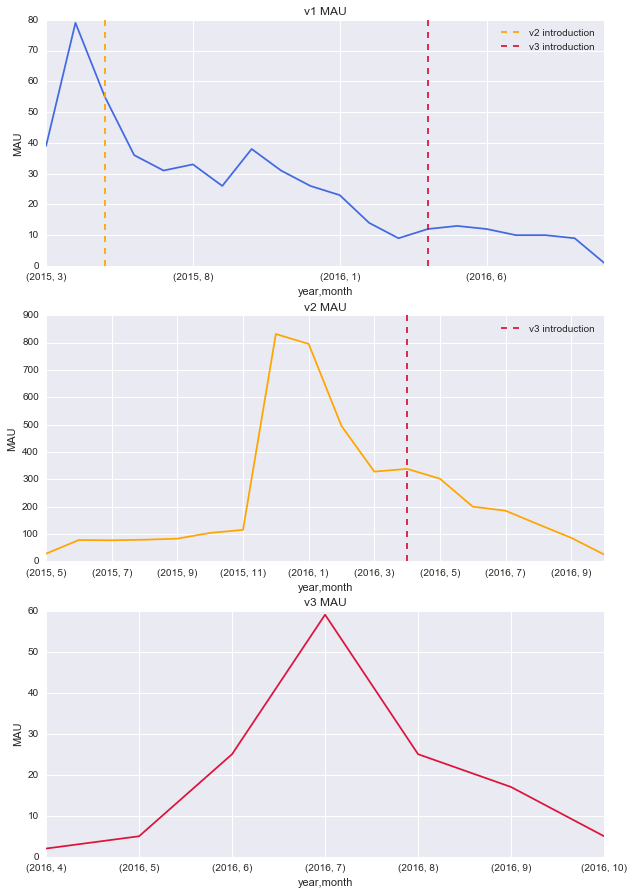


Fig.7 Application MAU by version

Fig.7 shows the MAU for different versions of the app. It can be seen that the record starts in March 2015, which is well after the first installation date recorded in the user data. Therefore the trend shown in v1 MAU might not tell the whole story. Nevertheless we can see MAU peaks at about 80 for v1 of the app. Shortly after that v2 was introduced and the MAU of v1 is on a steady decline. A decline was expected, as it is normal for some users to cease using the app as time goes by. It is also caused by the introduction of v2 so new installation (and use) of v1 would significantly slow down.

v2 started in May 2015 and the MAU gradually increased to over 100 in Nov 2015. Then a huge increase occurred in Dec 2015 reaching a new record high of over 800 active users per month. Then MAU started to decline sharply from Jan to Mar 2016. As v3 was introduced in Apr 2016, the MAU continued to drop to a number of under 50 in Oct 2016. v3 started in Apr 2016 and its MAU rises to a peak of about 60 in Jul 2016. However the number dropped continuously to below 10 in Oct 2016.

There are a few interesting insight from this MAU analysis and they are as follows:

1. There is an exceptional high peak near Dec 2015. It could be due to the fact that the app is related to products that are (can be) sold as Christmas presents (such as Kano's computer kits). Therefore when Christmas time came, the amount of users rose sharply. This can be further verified by doing a Daily Active Users (DAU) analysis for Dec 2015. If this is proved to be the case, we might want to also look at the data for Dec 2016. If the data shows a similar trend, the company can adjust it's sales/marketing/logistic strategy to better fulfill demand in the coming Christmas.

2. Comparing data between 2015 and 2016, it might be worthwhile to revisit the design/performance of v3. v3 was released in Apr 2016, arguably in a similar period when v2 was released back in 2015. However v3 hadn't demonstrated a similar amount of growth like v2. The growth in MAU stopped within 3 months with a significantly lower peak when compared to v2. Therefore it could be problem of the v3 app, or in a worse case the product is becoming less popular.

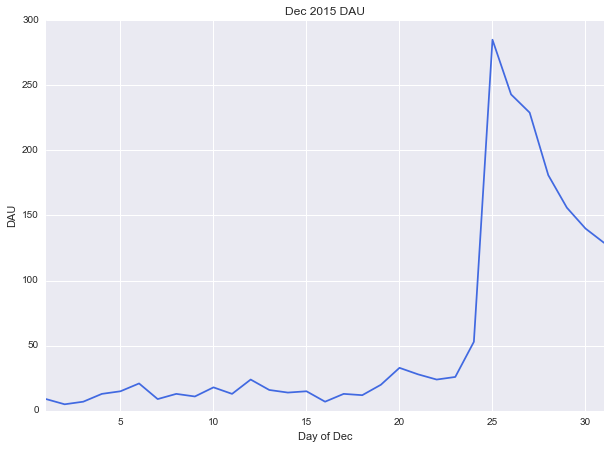


Fig.8 Daily Active User in Dec 2015

Fig.8 shows the daily active users for December 2015. As expected the peaking of active users was caused by Christmas. It can be seen that DAU peaked at 25th Dec and dropped off quickly towards the end of the month. This is not necessarily a bad thing though. It all depends on the company's strategy and how it makes its profit.

Retention

The definition of retention varies depending on the company's objective. In this case I would like to use the Rolling Retention as defined in [Applift](http://www.applift.com/blog/user-retention). The definition of this retention is the proportion of users coming back to the app on Day+N or any day after that. As we can see from the Fig.8, we ready know that not every user would come back everyday. Therefore it would be more appropriate to use a rolling retention to check whether users come back occasionally even after a certain amount of time, which is a hint that they are still interested in the product.

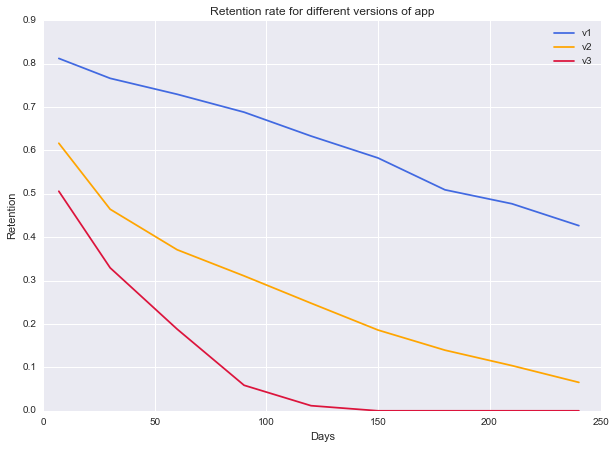


Fig.9 Retention curves

The above plot shows the retention rate for different versions of the app at different days after installation. It can be seen that the retention rate of v1 is the highest with over 50% of the customers continued using the app half a year. v2 seems to be significant worse in terms of rate. However we should bear in mind that a large number of users signed up (installed) the app during the Christmas period. It could be that the majority of them were not long term users hence left very soon after installation. This boosted the total number installed hence lowering the overall rate. The declining slope of v2 is actually quite similar to v1 hence further investigation has to be done in order to better compare v1 and v2. v3 however, seems to be underperforming. The slope of the retention curve is steeper and the app only managed to retain about half of its user 7 days since installation. Combining with other metrics such as total users and MAU, it can almost be certain that v3 is not performing particularly well.

Time spent in app

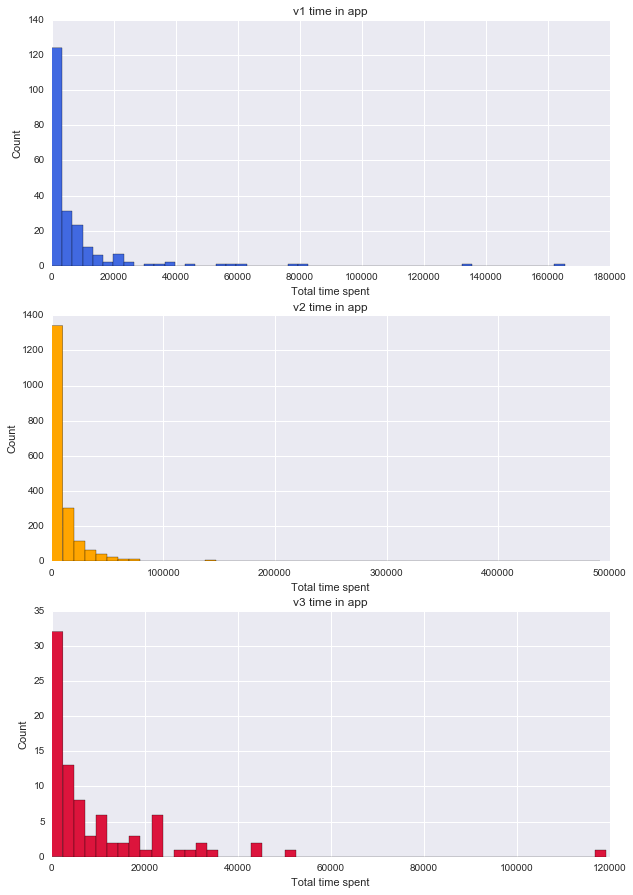
Time spent in app is a useful metric, which is used by analyst to understand the usage pattern of customers. It is very likely that the longer the user stay per session, the more engaged they were. From this data we can only tell the duration of sessions between events. There was no logging about users interaction hence it is not possible to tell what the users were doing when the app was running. Nevertheless this should still be an interesting metric to look at. Fig.10 shows the result.

Fig.10 Time in app for different versions

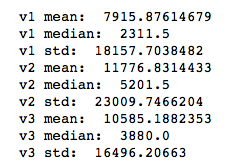


Fig.11 Time in app statistics

In order to quantify the difference in total time spent in app, permutation (randomization) tests were conducted. A permutation test is a hypothesis test that gives us an idea of whether the groups of data we have are significantly different. In terms of hypothesis tests, t-tests are more popular. However it is required that the data used in t-tests is normally distributed. It is obviously not the true in our case. Like the t-test, statistically significance can be determined by looking at the p-values. If the p-value were lower than the predetermined alpha level (0.05 is a commonly used figure), the difference would say to be significant.

From the values obtained, it can be seen that the p-value for the test between v1 and v2 is 0.018. This means that in terms of total time spent in app, v2 is significantly different from v1. However other statistics show values that are higher than 0.05 hence cannot be proved to be significant. One thing that has to be noticed is that from the results between v2, v3 and v1, v3, it seems that they are all not significantly different. However as mentioned above the result between v1, v2 has suggested otherwise. This could require more tests to be done to further confirm their relationship.

Fig.11 shows the statistics of the time in app for the different versions while Fig.12 shows the p-value calculated for each version pair.

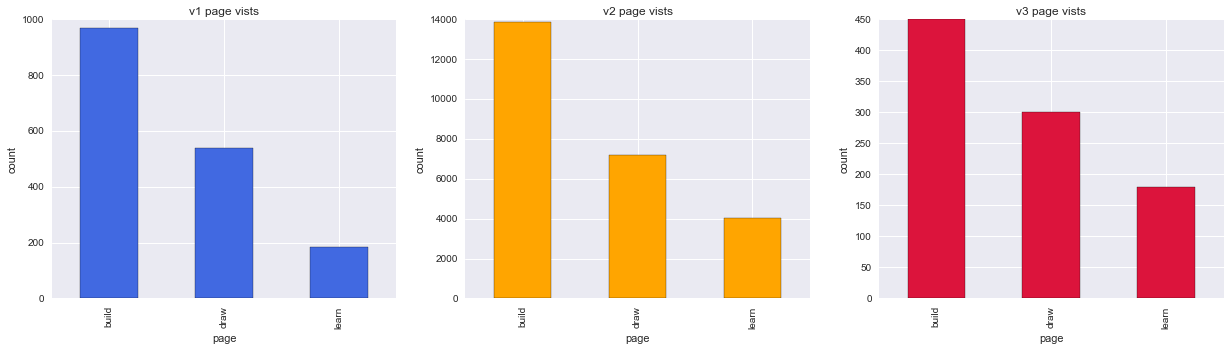


Fig.12 Results for permutation tests

Users behavior in different pages

From the preliminary analysis we saw that some of the sessions only lasted for a few seconds, which could represent non-meaningful (purposeful) interactions. Therefore it is decided to set a threshold time and below which the session would be considered not-interested. Without the knowledge of what the individual pages does, the threshold would need to be set by common sense and rationality. Recalling previous app using experiences, 30 seconds seems to be a reasonable starting point.

Having filtered out the data, bar charts were produced to measure and compare visit counts and visit time for the different pages in the 3 separate versions. These plots were shown in Fig.13. Fig.13 shows a very interesting finding. It can be seen that although the learning page has the lowest amount of visits across all versions, users tend to stay the longest there. It either means that the page itself requires a long staying time for users to achieve its purpose (the name 'learn' could imply its some kind of instructions) or it could mean that the users who ended up in this section of the app were really enjoying it.



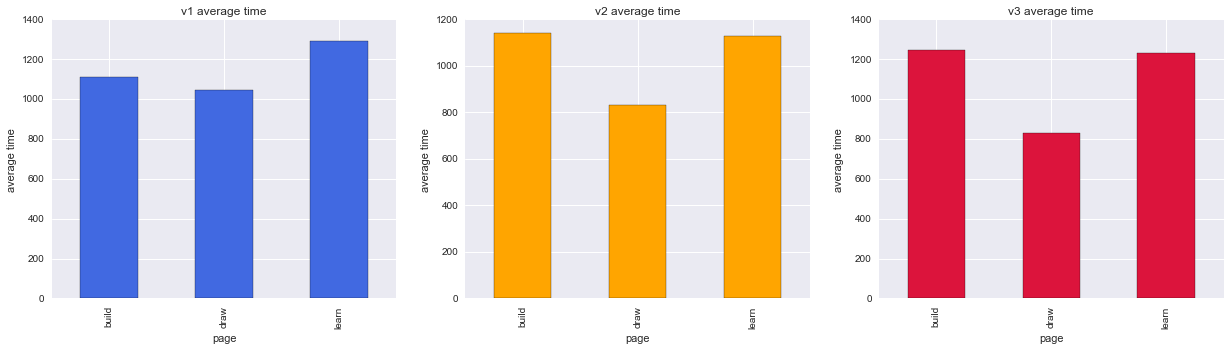


Fig.13 Access counts and average time in page for different pages and versions

Since a relatively small amount of users tend to spend a long time in the "learn" page, it is suspected that those are the enthusiastic users. Therefore it would be interesting to see whether those users spend more time on other pages too. Having plotted a heat map and scatter plot, it seems that the time spent in the pages are not that correlated. The scatter plot showed that for users who had visited both the "learn" and "build" page, only a few of them have spent over 50000 seconds in the "learn" page whilst it is much more common for the "build" page. In fact the difference in median value is over 3000 seconds (50 minutes).

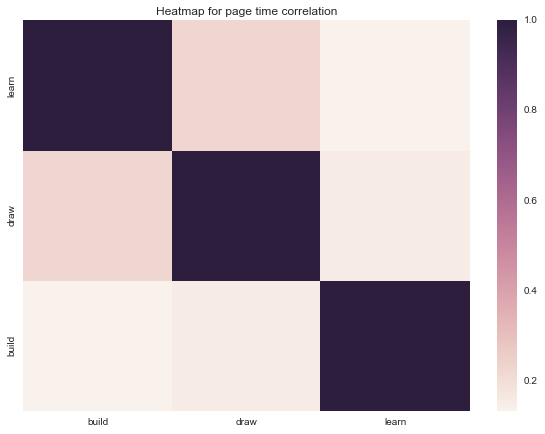
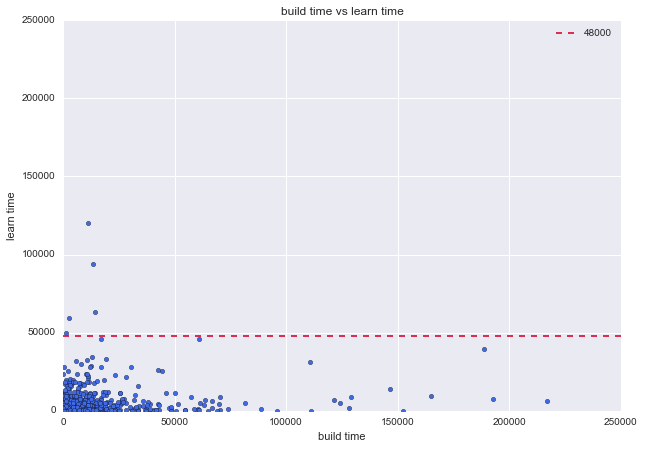


Fig.14 Heat map and scatter plots to demonstrate relationships

Conclusion

In this analysis we have looked at many different aspects of the available data. It can be said that the data is relatively clean and little wrangling was required. The only aspect that could be trickier would be the unknown app version record with some of the users. However considering that those instances accounted for a small part of the whole data set, it is decided to ignore them for the time being.

A few important KPIs were calculated and visualised using different kinds of plotting techniques. Total number of users usually increases steadily and the user accumulation ability for v1 and v2 were similar. However the performance of v3 seems to be worse comparatively. This was reflected in the shallower gradient shown in the plot. It is also interesting to discover that there were 2 time periods that saw rapid users growth. They were in December of 2014 and 2015 respectively.

Monthly active users analysis was telling a similar story with a large peak occurring at December 2015. This was reflected in the MAU plots for both the total and v2 (v2 was the most updated release at the time) data. In order to further investigate the peak, a daily active user for December 2015 was carried out which revealed that the peak occurred during Christmas time. Given that the installation data told a similar story the product owner can consider adjusting his sales and marketing strategy to even better capture this market. For the comparison between versions, v3 seems to be the weakest again. MAU only grew for 3 months before it started to decrease. The peak was also lower than the previous 2 versions.

Then retention data was also looked at. Rolling retention was used to determine whether users would come back after a certain amount of time. It is discovered that as the time gets longer, less and less user come back (which is perfectly reasonable). Again by carrying out comparisons between versions, v3 had demonstrated a weaker performance. It’s retention rate was the lowest for all periods and the drop in retention rate was larger than the other versions. It can be seen that v2 had an overall lower curve when comparing to v1. However considering v2 had a lot of “short term” users signing up during the Christmas peak, it is natural to have a lower overall retention rate. It can be seen that the change in retention rate was similar to v1, which suggests that the long-term customer retaining capability between those versions are similar.

It is safe to say that most customers spent less than 5.5 hours (20000 seconds ) in total in the app. Although the shape of the distribution was similar between the versions, the statistics could be different. Therefore a permutation test was carried out and shown that the time spent in app for v1 and v2 are significantly different. And by looking at the data again it could suggest that people spend more time overall in v2. However it should be noted that the result between v1, v3 and v2, v3 was said to be insignificant. Hence there is a possible conflict where further investigation would be required.

A brief user behavior analysis was carried out to gain more insight regarding the different areas in the app. It is discovered that despite being the least visited, the “learn” page had the highest average time spent. If learn was not an essential page (such as instructions), it could mean that the content in the page fits well with a niche group of users. It would be worth further investigating to possibly open up a new market. However in terms of relationship between pages, there seem to be very weak relationship. Using scatter plots and heat maps, it is revealed that users spending more time in one page do not necessarily spend more time in others. This could mean that in further development, it could be more efficient to get users to absolutely love the pages they were interested rather than trying to make a set of pages that felt mediocre for everybody.

Last but not least, the above analysis had only discovered a small portion of the available insights. Different types of investigations could be done to further make use of the available data. Examples include:

* Most popular time period for customer to use the app. (Can further be grouped by pages)
* User group segmentation (long term/short term) and their relative properties

For easier analysis in the future, it is also suggested that when new versions were released, the outdated version should immediately be made unavailable. This would eliminate the overlap in time to make version imputation a lot easier when necessary. Further more a better control of sample sizes would benefit the analysis. If new versions were released solely for trial purposes, the number of downloads should be limited in order to make comparison easier. For example it would be quite challenging to compare v2 and v3 in this analysis with sample sizes are very different.