Exploratory Data Analysis:

The 'users' table contains no duplicate rows and almost all of its columns contain non-null values. Only the 'churn_date' column contains NaN values but this is because the user is still using the calling plan. This table displays 500 customers or the phone company that are either were or currently are subscribed to a phone plan.

The 'plans' table contains no duplicate rows and all of its columns contain non-null values. This table displays the two phone plans to compare against each other and what each of them contains.

The 'calls' table contains no duplicate rows and all of its columns contain non-null values. This table displays all calls that a user made, including the data and duration of the call.

The 'messages' table contains no duplicate rows and all of its columns contain non-null values. This table displays all messages that a user sent, including the date it was sent.

The 'internet' table contains no duplicates and all of its columns contain non-null values. This table displays all internet sessions, including the dateusrs and the amount of mb used during the session.

Data Preprocessing:

In the 'users' table, I converted the 'reg_date' column to a DateTime Python object. I renamed the 'tariff' column to the 'plan' column.

In the 'calls' table, I converted the 'call_date' column to a DateTime Python object. I created a 'call_month' column by extracting only the numerical month value from the 'call_date' column. I renamed the 'id' column to 'call_id'. I decided to keep missed calls, whose 'duration' value is 0.0 because these calls won't affect the analysis. I rounded call duration upto a minute.

In the 'messages' table, I converted the 'message_date' column to a DateTime Python object. I created a 'message_month' column by extracting onlythe numerical month value from the 'message_date' column. I renamed the 'id' column to 'message_id'.

In the 'internet' table, I converted the 'session_date' column to a DateTime Python object. I created a 'session_month'column by extracting only the numerical month valuefrom the 'session_date' column. I renamed the 'id' column to 'internet_id'. I rounded internet session up to one megabyte.

In the 'plans' table, I converted the 'mb_per_month_included' column from megabytes to gigabytes by dividing by 1024. I renamed this column to 'gb_per_month_included'.

In the data, there are some irregularities where a client has entries for calls/messages/internet sessions passed their churn data (date they stopped using the service). For these cases, I will assume that this is a mistake and these rows are removed.

I sliced the 'users' table for users who have a non-null churn date and extracted their user_id / churn_date. For every user who has a non-null churn date, I looked at the calls, messages, internet tables for their entries and I dropped any row in these tables with a matching user_id and whose call/message/session_date is greater than their churn date.

To calculate the number of calls made and the minutes used per month for each user, I created a pivot table that calculates the number of calls made and the total sum of the duration of calls made for each user for each month. Such that: for each user and for each month, how many calls did they make that month and how many minutes did they use up that month. I reset the index of the table and reformatted column names for table readability.

To calculate the number of message sent per month for each user, I created a pivot table that calculates the amount of messages sent for each user for each month. Such that: for each user and for each month, how many messages did they sent that month. I reset the index of the table and reformatted column names for table readability.

	user_id	call_month	calls_made	minutes_used_per_month
0	1000	12	16	124.0
1	1001	8	27	182.0
2	1001	9	49	315.0
3	1001	10	65	393.0
4	1001	11	64	426.0
...	...	...	...	...
2216	1498	12	39	339.0
2217	1499	9	41	346.0
2218	1499	10	53	385.0
2219	1499	11	45	308.0
2220	1499	12	65	496.0

	user_id	message_month	messages_sent_per_month
0	1000	12	11
1	1001	8	30
2	1001	9	44
3	1001	10	53
4	1001	11	36
...	...	...	...
1772	1496	9	21
1773	1496	10	18
1774	1496	11	13
1775	1496	12	11
1776	1497	12	50

	user_id	session_month	gb_used_per_month
0	1000	12	2.0
1	1001	8	7.0
2	1001	9	14.0
3	1001	10	22.0
4	1001	11	19.0
...	...	...	...
2235	1498	12	23.0
2236	1499	9	13.0
2237	1499	10	20.0
2238	1499	11	17.0
2239	1499	12	22.0

To calculate the number of GBs used per month for each user, I created a pivot table that calculates the total sum of MB used for each user for each month. Such that: for each user and for each month, how many MB did they use up that month. Once that sum of MB was calculated, I converted it to GB. I reset the index of the table and reformatted column names for table readability.

To calculate the monthly profit from each user, I started by merging together the 3 pivot tables I created in the last step. Because I want to merge the tables by user AND by month (user 1 - month8, user1 - month9, etc), I completed an outer merge (so that all rows from both tables are included) with specifications on which columns from the left merging table and specifications on the right merging table. For example: when merging the userCalls and the userMessages table, I wanted to include the call_month column from the userCalls table and the message_month column from the userMessages table. I repeated the same when merging userInternet table and userPlans table into the mix.

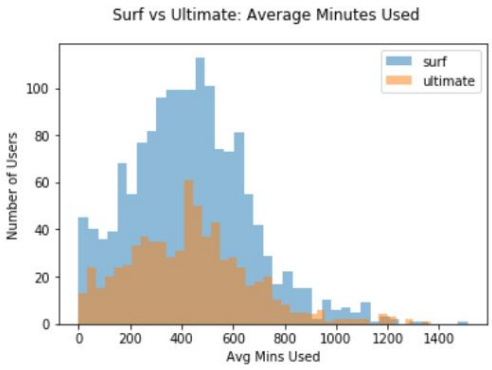
In the end, I assembled a userUsage table which: for each user and for each month contains how many messages sent that month, how many minutes used that month and how much GB of data used up that month.

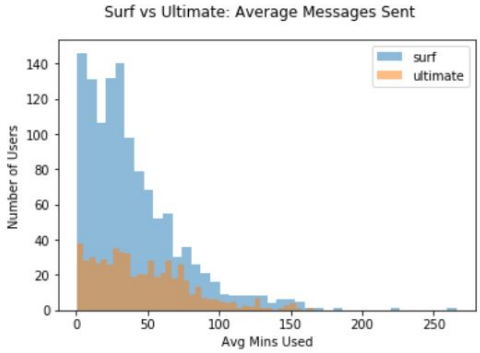
To add the monthly profit to this table, I wrote a function that goes through each row in this userUsage column, parses the necessary information and calculates the profit. For each row: it extracts the user_id, the minutes_used, messages_sent and gb_used. Then, it extracts the plan type that user subscribes to and extracts the package limits and the costs after the user surpasses the limits. I subtracted the free package limit from the amount the user used (for minutes, messages, gb), multiplied the results by the calling plan values, added up the additional charge values and then added the monthly charge to that value. The results was the profit the company made from that user that month given that plan and it was stored in the profit column of the userUsage table.

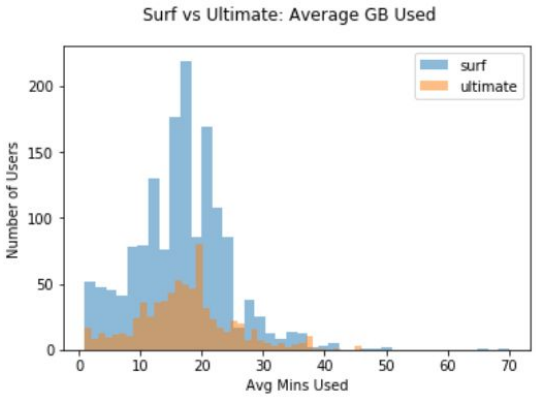
	user_id	call_month	minutes_used_per_month	message_month	messages_sent_per_month	session_month	gb_used_per_month	plan	month_profit
0	1000	12.0	124.0	12.0	11.0	12.0	2.0	ultimate	70.00
1	1001	8.0	182.0	8.0	30.0	8.0	7.0	surf	20.00
2	1001	9.0	315.0	9.0	44.0	9.0	14.0	surf	20.00
3	1001	10.0	393.0	10.0	53.0	10.0	22.0	surf	90.09
4	1001	11.0	426.0	11.0	36.0	11.0	19.0	surf	60.00
...	...	...	...	...	...	...	...	...	...
283	1349	NaN	NaN	12.0	61.0	NaN	NaN	surf	20.33
284	1349	NaN	NaN	NaN	NaN	10.0	13.0	surf	20.00
285	1349	NaN	NaN	NaN	NaN	11.0	17.0	surf	40.00
286	1349	NaN	NaN	NaN	NaN	12.0	13.0	surf	20.00
287	1108	NaN	NaN	NaN	NaN	12.0	1.0	ultimate	70.00

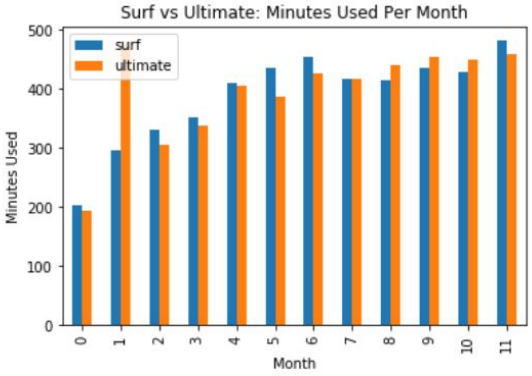
To find the minutes, messages and GB the users of each plan require per month, I calculated the mean, dispersion (variance) and standard deviation for surf (minutes used, messages sent, GB used) and ultimate (minutes used, messages sent, GB used).

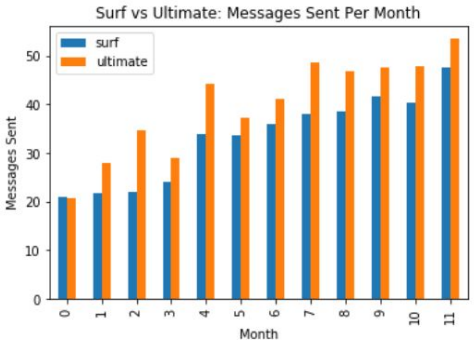
The x axis shows the number of minutes/messages/gigabytes, and the y axis shows the number of users who use that many minutes/messages/gigabytes a month.

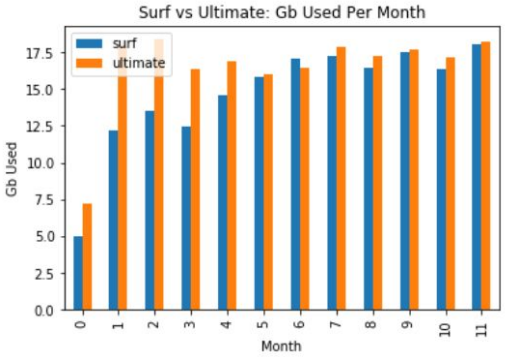












We want to test the hypothesis that the average profit from users of ultimate and surf plans differ. We can do this using an independent samples t-test to compare the means from two groups (ultimate and surf users). We can apply a t-test here due to the Central Limit Theorem, which implies that you can estimate the mean of a statistical population using the mean of a sample, and since the means are approximately normally distributed, we can use the t-test

Null Hypothesis H0: The average profit from users of Ultimate does not differ from the average profit from users of Surf.
Alternative Hypothesis H1: The average profit from users of Ultimate does differ from the average profit from users of Surf.

The p_value tells us that we can reject the null hypothesis, thus indicating that the average profit of Ultimate users does differ from the average profit of Surf users. This can be further proven by looking at the numbers: The average monthly profit from ultimate users is 72.34 while the average monthly profit from surf users is 53.6.

We want to test the hypothesis that the average profit from users in NY-NJ area is different than that of users from other regions. We can do this using an independent samples t-test to compare the means from two groups (NY-NJ and non NY-NJ users). We can apply a t-test here due to the Central Limit Theorem, which implies that you can estimate the mean of a statistical population using the mean of a sample, and since the means are approximately normally distributed, we can use the t-test

Null Hypothesis H0: The average profit from users in NY-NJ does not differ from the average profit from users of other regions.
Alternative Hypothesis H1: The average profit from users in NY-NJ does differ from the average profit from users of other regions.

The p_value tells us that we can reject the null hypothesis, thus indicating that the average profit of NY-NJ users does differ from the average profit of non NY-NJ users. This can be further proven by looking at the numbers: The average monthly profit from NY-NJ users is 56.10 while the average monthly profit from non NY-NJ users is 60.22.

Overall Conclusion:

I found that the difference between the profit made from Ultimate and Surf was statistically significant enough to conclude that the Ultimate plan is more profitable than the Surf plan and I recommend that the commercial department adjust the advertising budget such that the Ultimate plan receives more advertisements.

The average monthly profit made from surf users is 53.6 while the avergae monthly profit made from Ultimate users if 72.34.