Prediction.io - plots

Documentation

Gabriela Kaczka

number_of_new_users_per_month()

Data

Tables	Columns
Users	'signupTime', 'userId'

Properties

dropped 'Nan' and 'None' values from 'signupTime' in Users

Actions

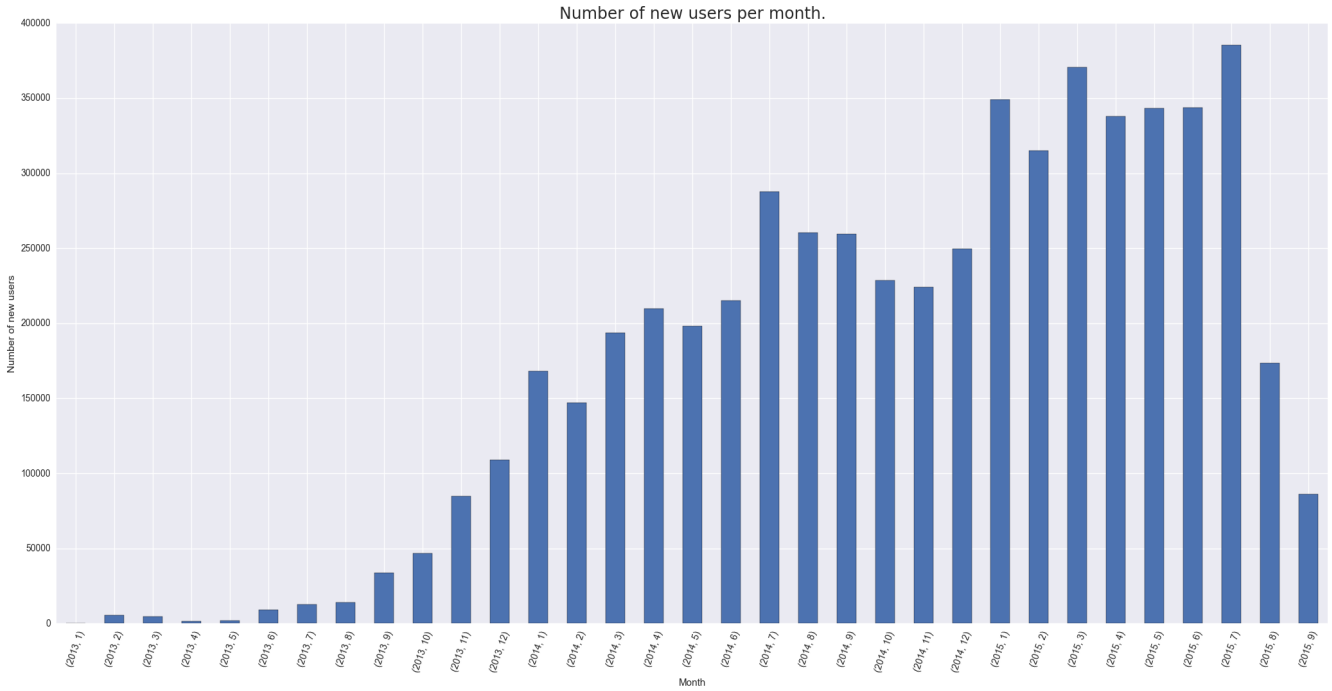
performing count() operation on 'userId', grouped by year and month in 'signupTime'

sorting 'signupTime' by year and month

Axes

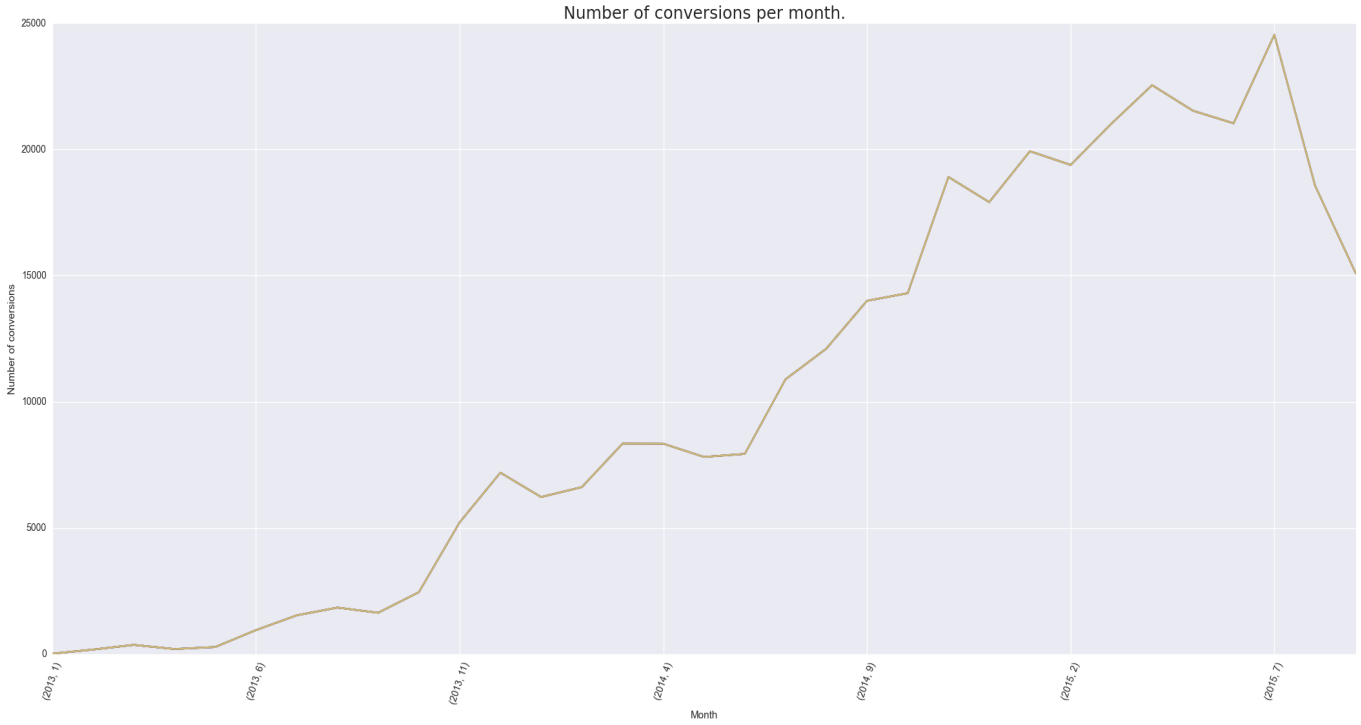
x: year and month

y: number of registrations



As can be seen from plot, the number of new users was constantly increasing since beginning of the site till July 2015, then, as plot shows, was a rapid crash in the number of newly registered people.

number_of_conversions_per_month()	
Data	
Tables	**Columns**
Conversions	'timestamp'
Properties	
dropped 'Nan' and 'None' values from 'timestamp' in Conversions	
Actions	
performing count() operation on rows grouped by year and month	
sorting 'timestamp' by year and month	
Axes	
x: year and month	
y: number of conversions	



As plot indicates the number of conversions per month reached a peak in July 2015, then began to decrease.

number_of_items_purchased_per_month()

Data

Tables	Columns
Conversions	'timestamp', 'quantity'

Properties

dropped 'Nan' and 'None' values from 'timestamp' and 'quantity' in Conversions

Actions

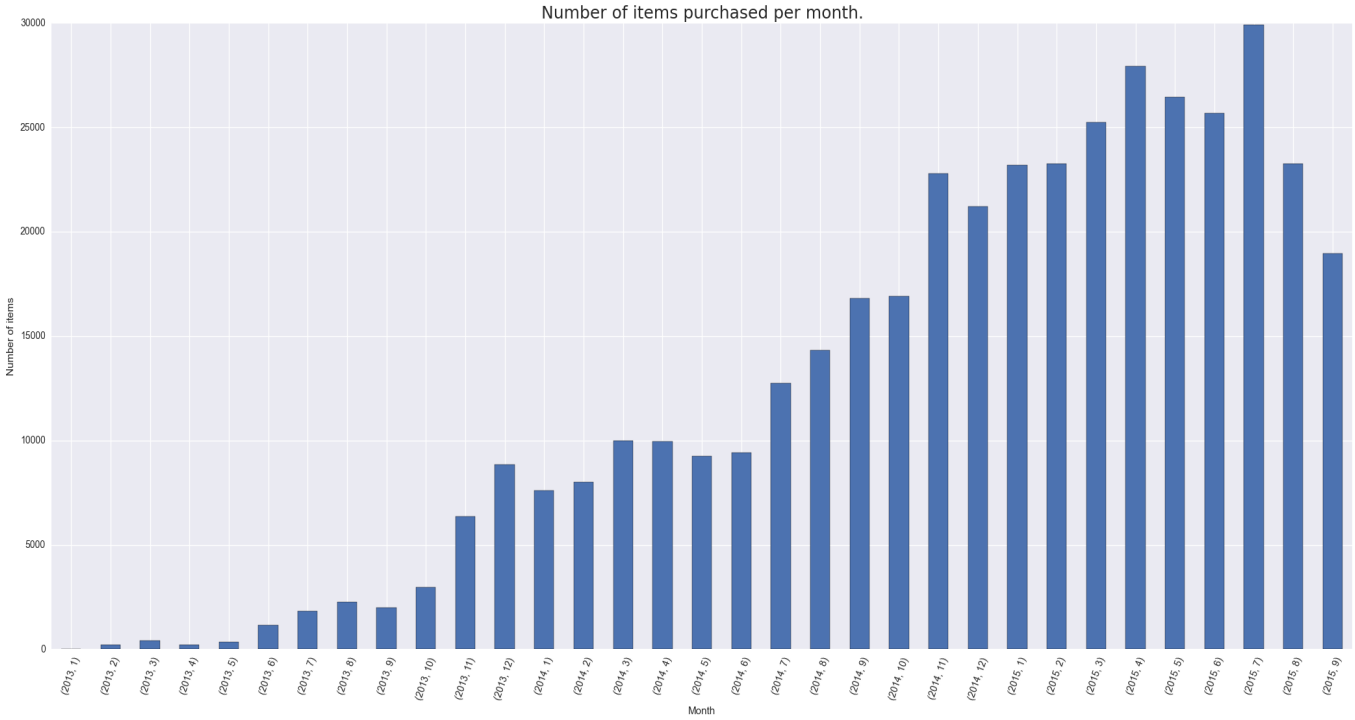
performing sum() operation on 'quantity', rows grouped by year and month of conversion

sorting 'timestamp' by year and month

Axes

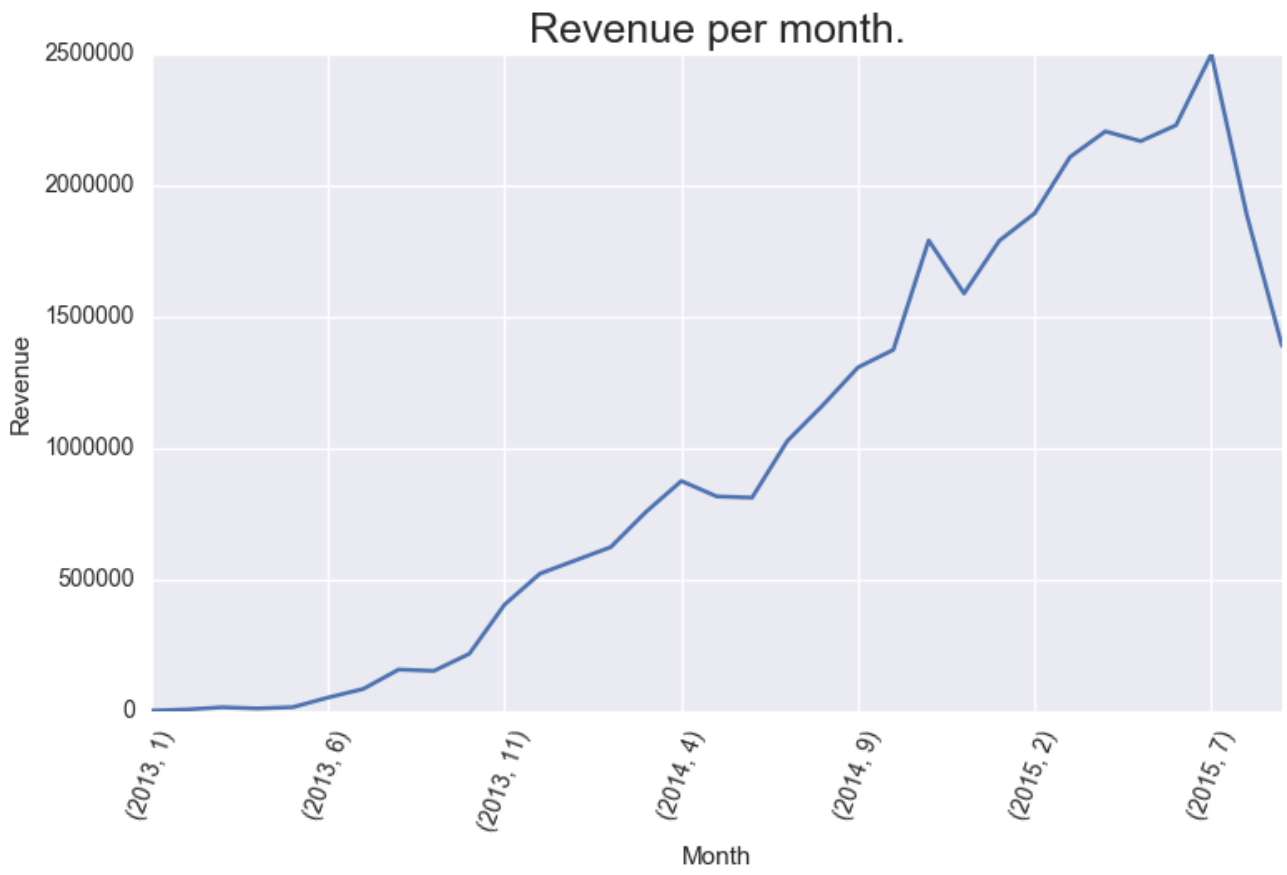
x: year and month

y: number of purchased items



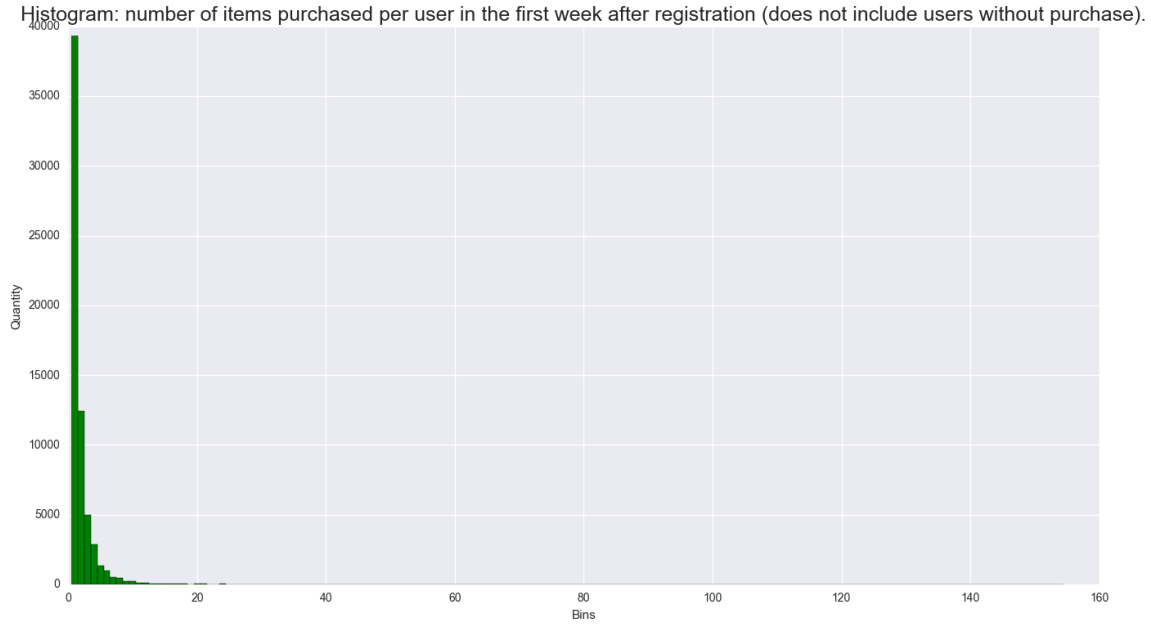
As can be seen, again, the peak is reached in July 2015, then decrease in purchases appeared.

revenue_per_month()	
Data	
Tables	**Columns**
Conversions	'timestamp', 'price'
Properties	
dropped 'Nan' and 'None' values from 'timestamp' and 'price' in Conversions	
Actions	
performing sum() operation on 'price', rows grouped by year and month of conversion	
sorting 'timestamp' by year and month	
Axes	
x: year and month	
y: income	



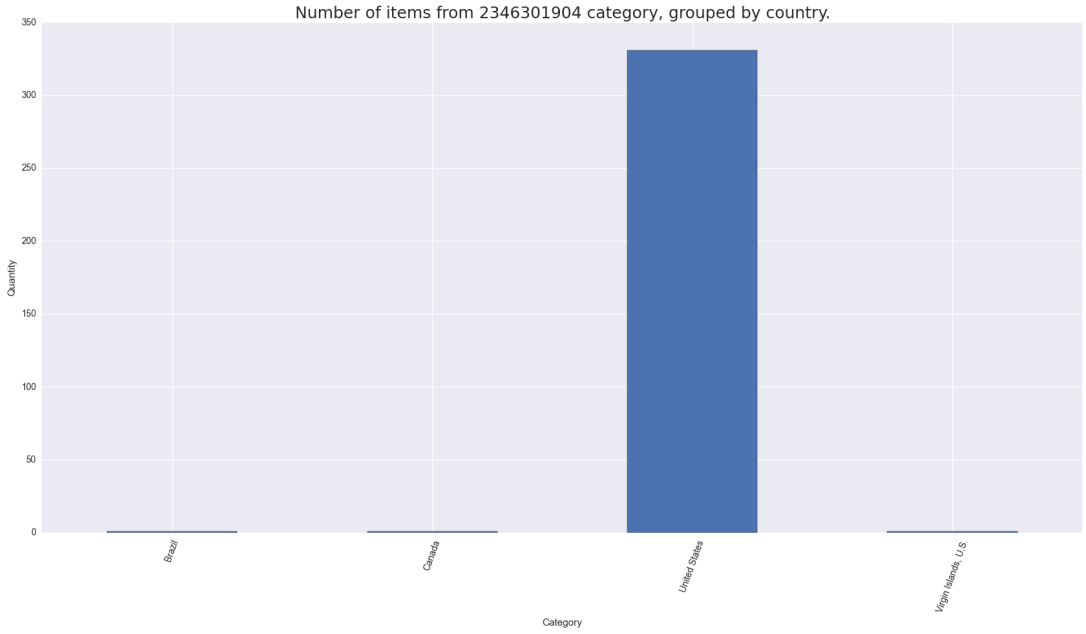
As plots shows among 2013, 2014 and 2015 income was constantly growing. Interesting points might be seen in November 2014 and July 2015.

number_of_items_purchased_per_user_in_the_first_week_after_registration_histogram()	
Data	
Tables	**Columns**
Conversions	'timestamp', 'userId', 'quantity'
Users	'signupTime', 'userId'
Properties	
dropped 'Nan' and 'None' values from 'timestamp', 'userId', 'quantity' in Conversions	
dropped 'Nan' and 'None' values from 'signupTime', 'userId' in Users	
doesn't include information about users, who haven't got any purchase	
Actions	
joining Conversions and Users on 'userId'	
adding additional column to joined structure: 'week_after' - date week after registration	
filtering 'timestamp' - rows only with 'timestamp' <= 'week_after' preserved	
performing sum() operation on 'quantity', rows grouped by 'userId'	
bins in range(1, max(grouped.values)+2), every 10th bin preserved	
Axes	
x: bins	
y: quantity	



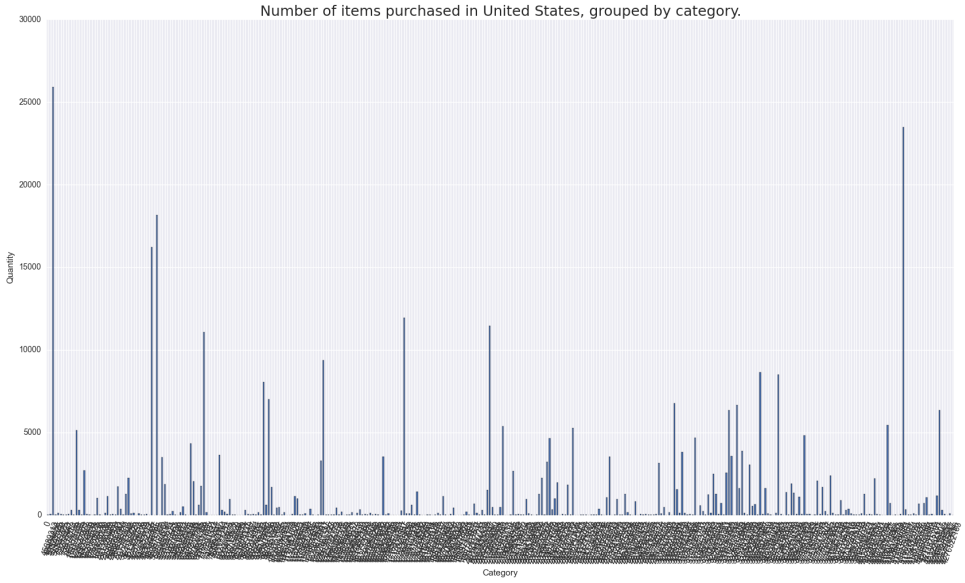
The plot indicates that majority of people with at least one purchase during first week after sign up decide not to buy more than one thing.

number_of_items_purchased_from_particular_category_grouped_by_country(category)	
Data	
Tables	**Columns**
Conversions	'itemId', 'userId', 'quantity'
Items	'itemId', 'category'
Users	'userId', 'registerCountry'
Properties	
dropped 'Nan' and 'None' values from 'category' in Items	
dropped 'Nan' and 'None' values from 'quantity' in Conversions	
dropped 'Nan' and 'None' values from 'registerCountry' in Users	
Actions	
joining Items and Conversions on 'itemId' and futher with Users on 'userId'	
filtering joined data on 'category' property	
performing sum on 'quantity' in rows grouped by 'registerCountry'	
Axes	
x: country	
y: quantity	



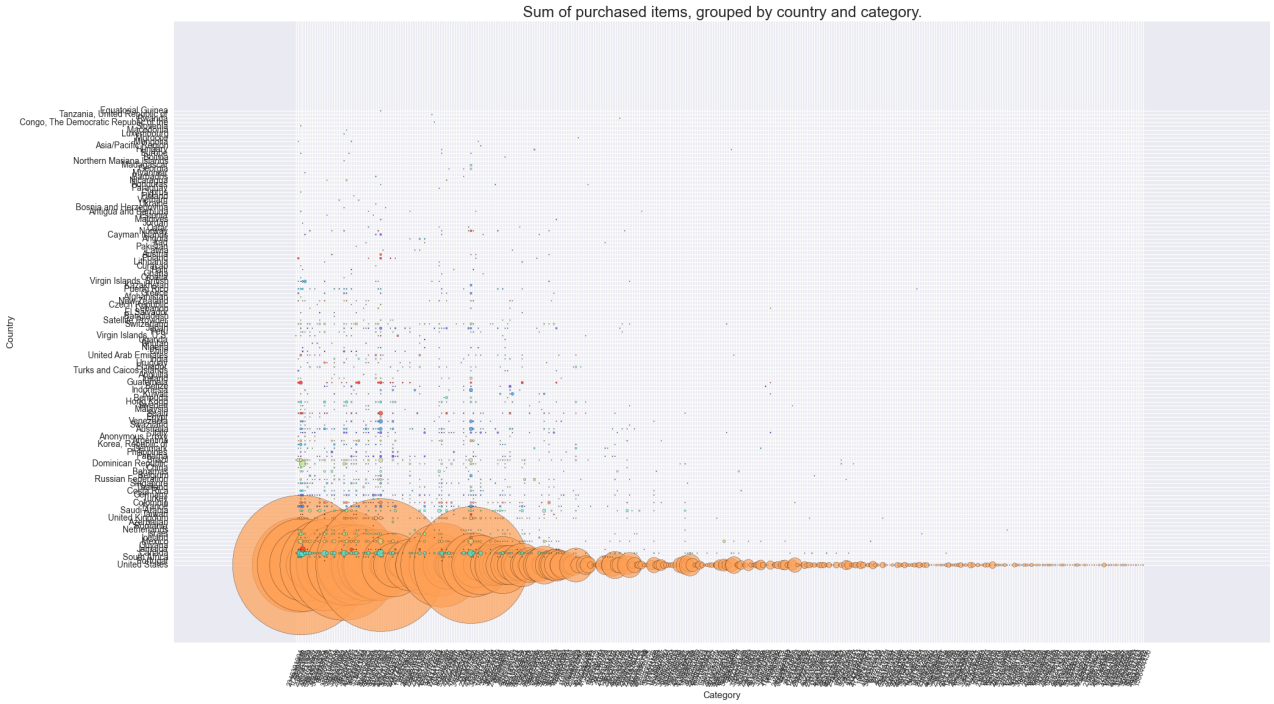
Above plot is generated with filter category == 2346301904. However one plot is not representative enough, what might be seen among figures is that number of purchases in United States is the biggest.

number_of_items_purchased_in_particular_country_grouped_by_category(country)	
Data	
Tables	**Columns**
Conversions	'itemId', 'userId', 'quantity'
Items	'itemId', 'category'
Users	'userId', 'registerCountry'
Properties	
dropped 'Nan' and 'None' values from 'category' in Items	
dropped 'Nan' and 'None' values from 'quantity' in Conversions	
dropped 'Nan' and 'None' values from 'registerCountry' in Users	
Actions	
joining Items and Conversions on 'itemId' and futher with Users on 'userId'	
filtering joined data on 'country' property	
performing sum on 'quantity' in rows grouped by 'category'	
Axes	
x: category	
y: quantity	



Above plot is an example generated for United States. It indicates that some categories are extremely popular, whereas purchases in the others are on similar level.

number_of_purchased_items_grouped_by_categories_in_all_countries()	
Data	
Tables	**Columns**
Conversions	'itemId', 'userId', 'quantity'
Items	'itemId', 'category'
Users	'userId', 'registerCountry'
Properties	
dropped 'Nan' and 'None' values from 'category' in Items	
dropped 'Nan' and 'None' values from 'quantity' in Conversions	
dropped 'Nan' and 'None' values from 'registerCountry' in Users	
Actions	
joining Items and Conversions on 'itemId' and further with Users on 'userId'	
generating y-axis' ticks on unique 'registerCountry' values	
generating x-axis' ticks on unique 'category' values	
filtering data on 'registercountry' and 'category' property	
performing sum() operation on 'quantity' in filtered rows	
setting ticks and labels on the plot	
Axes	
x: category	
y: country	



As can be seen United States are extremely important client for the service. The others countries whose impact in total amount of purchased products is significant are: Dominican Republic, Guatemala, Spain, Venezuela Canada and Mexico.

number_of_purchased_items_after_seeing_campaigns_grouped_by_categories()

Data

Tables	Columns
Conversions	'itemId', 'userId', 'quantity'
Users_Ads	'itemId', 'category'
Users	'userId', 'registerCountry'
Items	

Properties

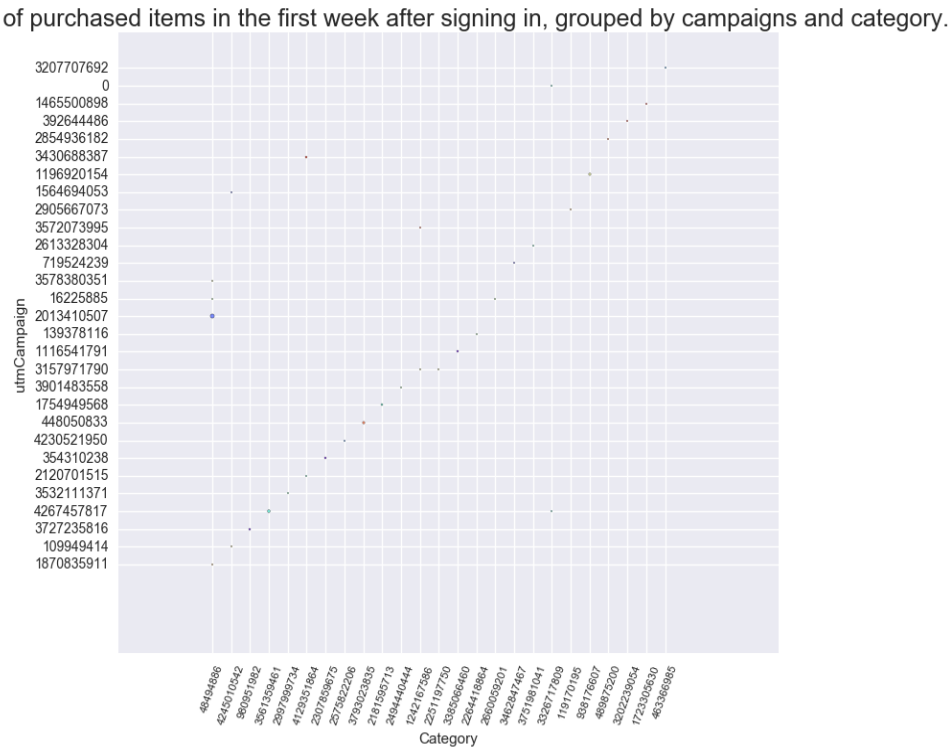
dropped 'Nan' and 'None' values from 'timestamp' in Conversions
dropped 'Nan' and 'None' values from 'signupTime' in Users
dropped 'Nan' and 'None' values from 'utmCampaign' in Users_Ads
dropped 'Nan' and 'None' values from 'category' in Items

Actions

adding additional column to Users: 'week_after' - date week after registration
joining Users, Users_ads and Conversions on 'userId' and further with Items on 'itemId'
filtering joined structure: joined['timestamp'] <= joined['week_after']
generating y-axis' ticks on unique 'utmCampaign' values
generating x-axis' ticks on unique 'category' values
filtering data on 'utmCampaign' property
performing sum() operation on 'quantity' in filtered rows
setting ticks and labels on the plot

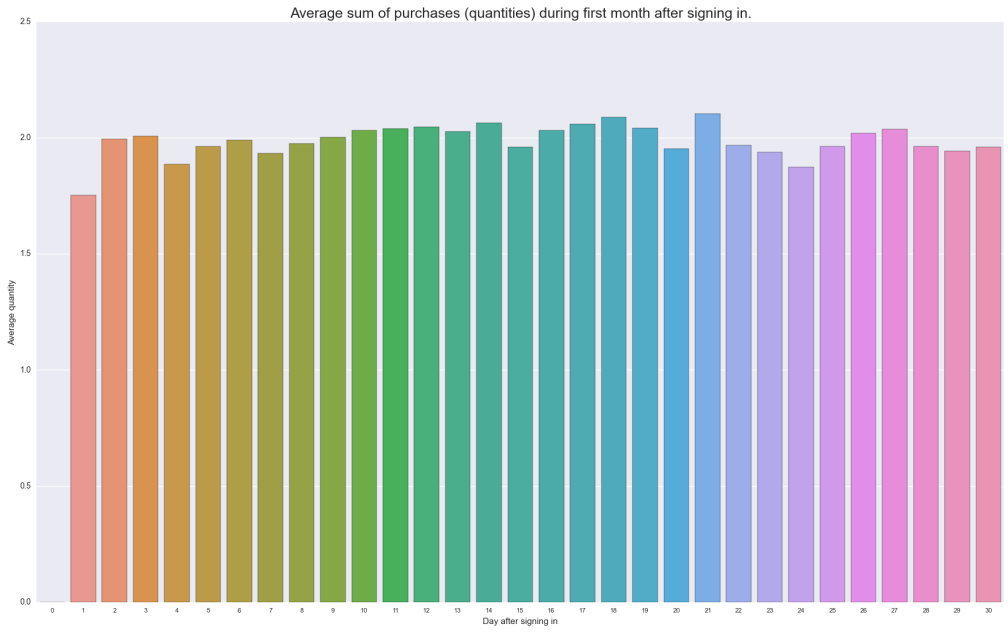
Axes

x: category
y: utmCampaign



Above plot is generated on a random sample (0.15) of the data.

average_number_of_purchased_items_during_the_first_month_after_signing_in()	
Data	
Tables	**Columns**
Conversions	'userId', 'timestamp', 'quantity'
Users	'userId', 'signupTime'
Properties	
dropped 'Nan' and 'None' values from 'timestamp' and 'quantity' in Conversions	
dropped 'Nan' and 'None' values from 'signupTime' in Users	
doesn't include information about users who haven't any purchase in particular day	
Actions	
joining Users and Conversions on 'userId'	
adding additional column: 'purchase_day' – number of days after 'signupTime', when conversion was completed	
filtering joined structure: 0<='purchase day'<30	
grouping rows on 'purchase_day' and 'userId' (one user can have many conversions during one day)	
performing sum() operation on grouped structure, a result is number of purchased items in particular day after registration	
counting average: for each day in the range (0, 30]	
Axes	
x: day after registration	
y: average quantity	



As can be

seen, the plot is monotonic, the average number of purchased items per day during first month after registration, oscillates around 2.

histogram_number_of_purchases_per_user_during_first_week_and_month()	
Data	
Tables	**Columns**
Conversions	'userId', 'timestamp', 'quantity'
Users	'userId', 'signupTime'
Properties	
dropped 'Nan' and 'None' values from 'timestamp' and 'quantity' in Conversions	
dropped 'Nan' and 'None' values from 'signupTime' in Users	
this plot take into account users without purchase in established periods	
interactive plot: https://plot.ly/~PythonAPI/272.embed	
Actions	
adding additional column: 'week_after' – date week after 'signupTime' - to Users	
adding additional column: 'month_after' – date month after 'signupTime' - to Users	
joining Users and Conversions on 'userId'	
filtering joined structure: 'timestamp' <= 'week_after' and 'timestamp' <= 'month_after' respectively	
grouping rows on 'userId' and performing sum() operation on 'quantity' on grouped structures (the result is number of purchased items per user during first week and month after sign up)	
Axes	
x: bins	
y: quantity	

As can be seen almost 20k of people decided to not buy anything in the first week after registiartion, however decided to purchase product during next three weeks.

probability_of_purchase_during_the_first_month_after_registration()

Data

Tables	Columns
Conversions	'userId', 'timestamp', 'quantity'
Users	'userId', 'signupTime'

Properties

dropped 'Nan' and 'None' values from 'timestamp' and 'quantity' in Conversions

dropped 'Nan' and 'None' values from 'signupTime' in Users

Actions

adding additional column: 'purchase_day' – number of days after 'signupTime' to Users

joining Users and Conversions on 'userId'

obtaining the number of all users

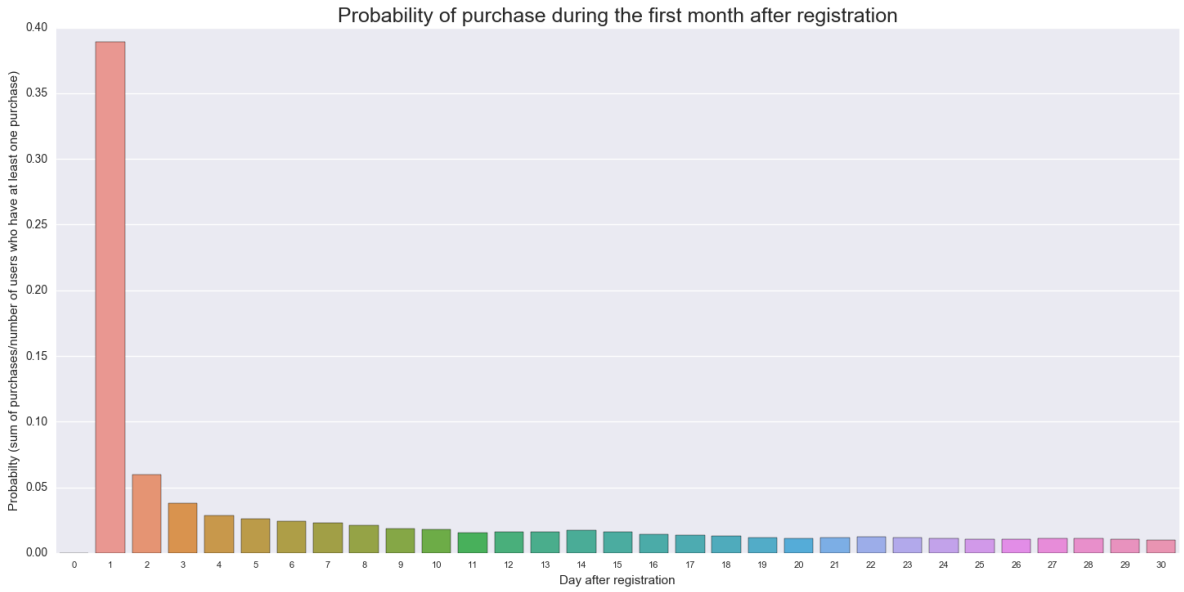
filtering joined structure: 0 <= 'purchase_day' < 30

grouping rows on 'purchase_day' and performing count() operation on 'userId' on grouped structure (for each day in range (0,30) the result is number of users who purchased at least one item in this day)

to count probability of purchase each value in purch is divided by number of all users

Axes

x: day after registration



y: probability (sum of purchases/number of users who have at least one purchase)

Above plot indicates that it is most likely that users will buy something in the first few days after registration.

h_pd_igd_weekly_user_count_of_purchases()

Data

Tables	Columns
Conversions	'timestamp', 'quantity'

Properties

dropped 'Nan' and 'None' values from 'timestamp' and 'quantity' in Conversions

Actions

adding additional columns: 'week' (week in year of conversion) and 'year' (year of conversion) to Conversions

grouping rows on 'year' and 'week value'

performing sum on 'quantity' column on grouped structure

displaying histogram of the data

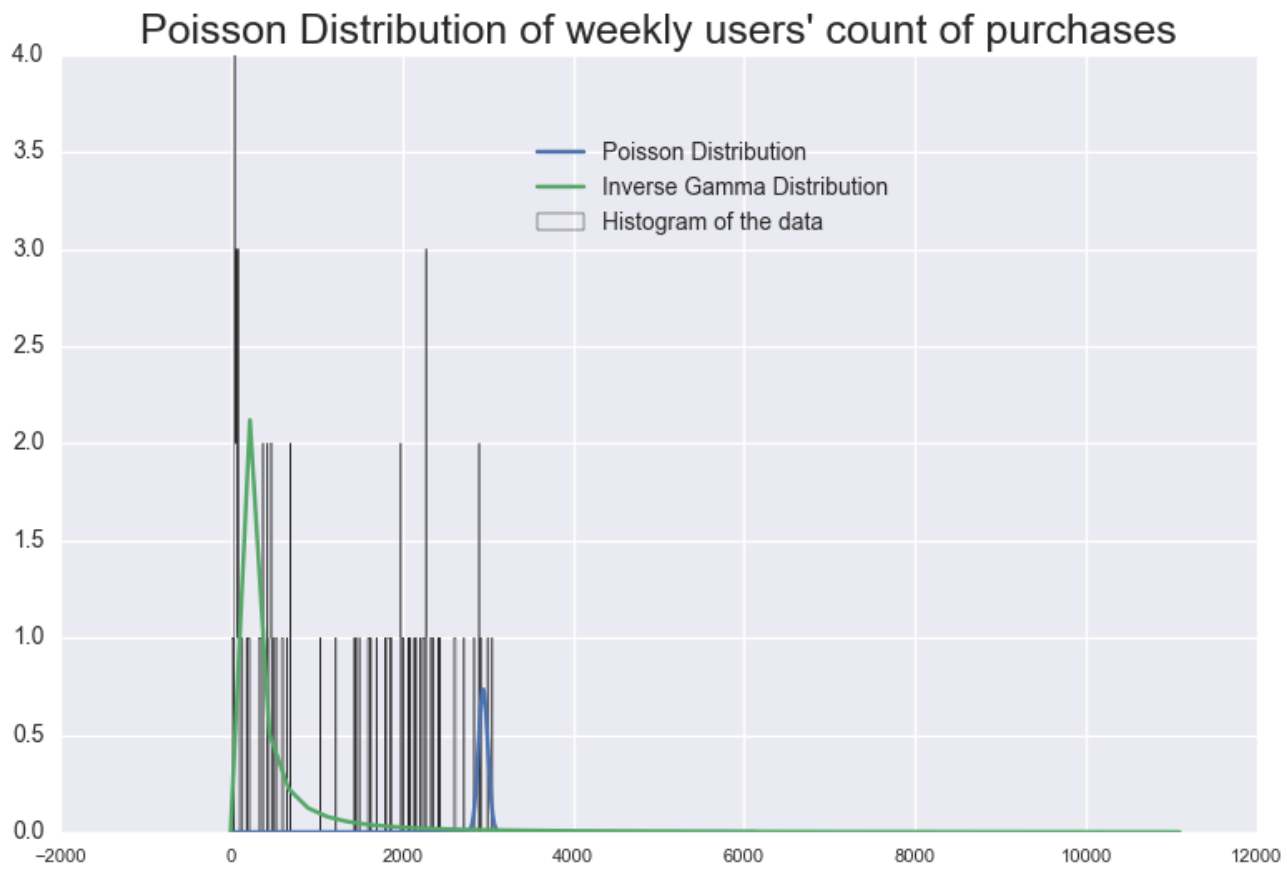
displaying Poisson Distribution of the data

displaying Inverse Gamma Distribution of the data

Axes

x: day after registration

y: probability (sum of purchases/number of users who have at least one purchase)



As above plots indicates weeks differ from each other in count of purchases.

number_of_active_users_per_month()

Data

Tables	Columns
Conversions	'timestamp'
Users	'userId'

Properties

dropped 'Nan' and 'None' values from 'timestamp' in Conversions

dropped 'Nan' and 'None' values from 'signupTime' and 'userId' in Users

Actions

adding additional columns: 'month' (month of conversion) and 'year' (year of conversion) to Conversions

joining Users and Conversions on 'userId'

counting number of active users per month and generating plot

adding additional columns: 'month' (month of registration) and 'year' (year of registration) to Users

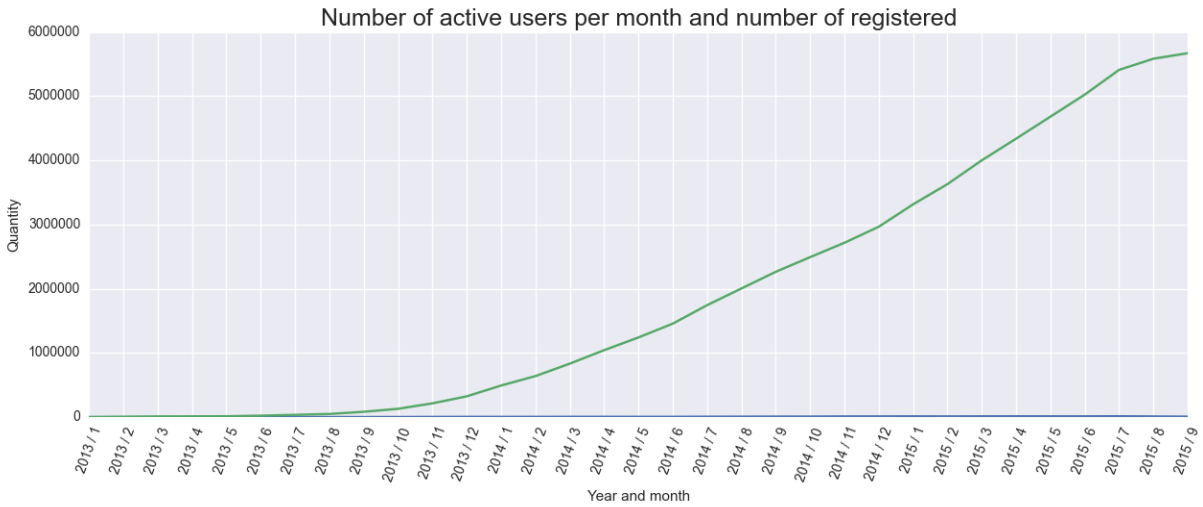
grouping Users by 'year' and 'month', performing count() operation on 'userId'

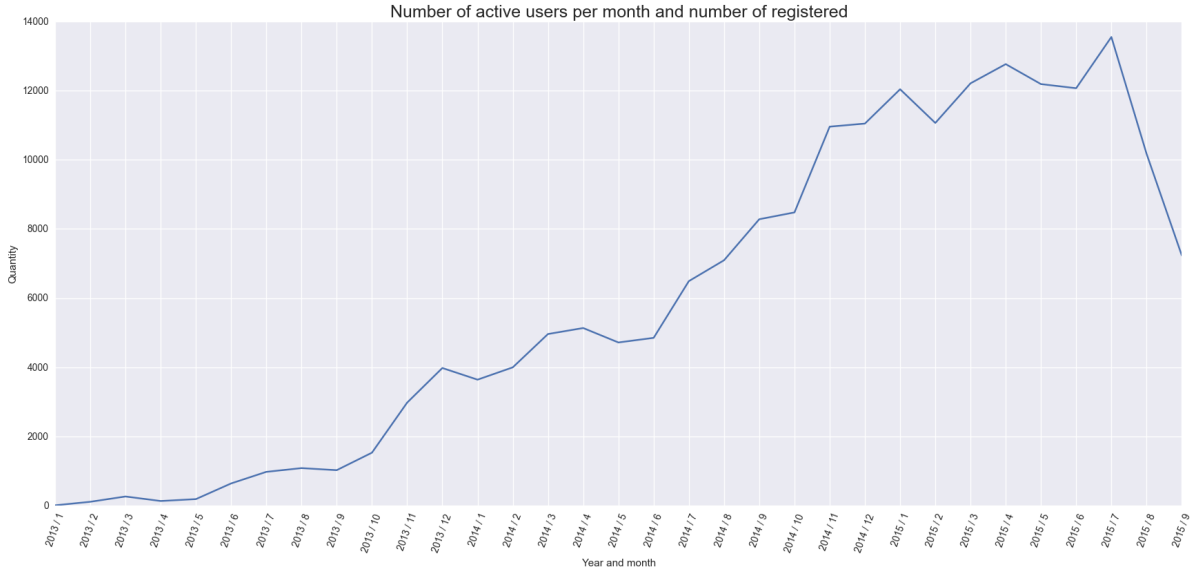
counting overall number of registered users and displaying plot

Axes

x: year and month

y: quantity





As above plots indicate number of active users (second plot) was quite proportional to number of registered users (first plot).

variance_sum_of_revenue_per_user_in_each_month()

Data

Tables	Columns
Conversions	'timestamp'
Users	'userId'

Properties

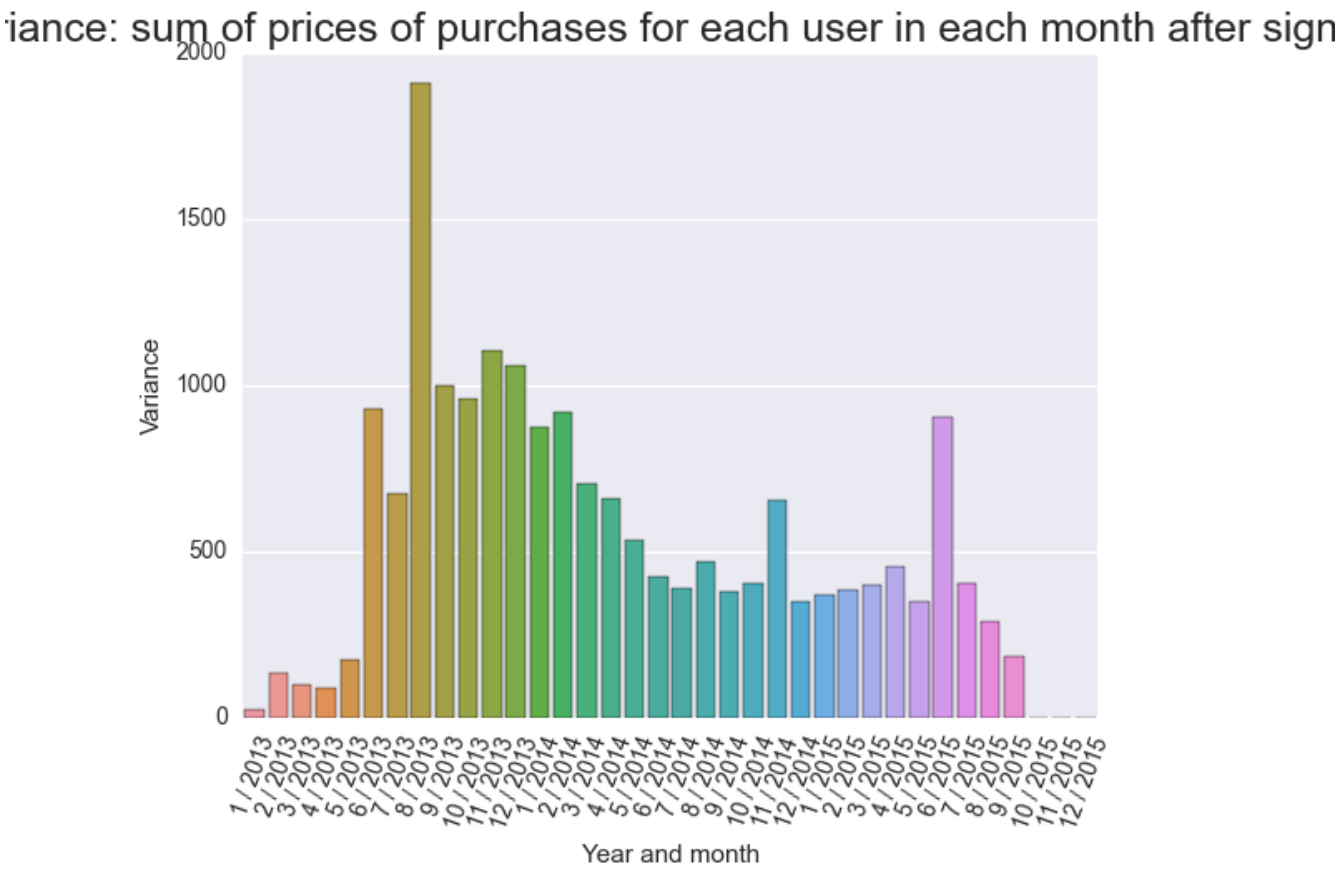
dropped 'Nan' and 'None' values from 'timestamp' in Conversions
dropped 'Nan' and 'None' values from 'userId' in Users

Actions

adding additional columns: 'conv_month' (month of conversion) and 'conv_year' (year of conversion) to Conversions
adding additional columns: 'signup_month' (month of registration) and 'signup_year' (year of registration) to Users
joining Users and Conversions on 'userId'
grouping joined structure on 'conv_year', 'conv_month' and 'userId', performing sum() operation on 'price'
performing second groupby()on 'signup_year' and 'signup_month', and then performing count() operation to obtain number of signed users
counting variance of revenue per user for each month

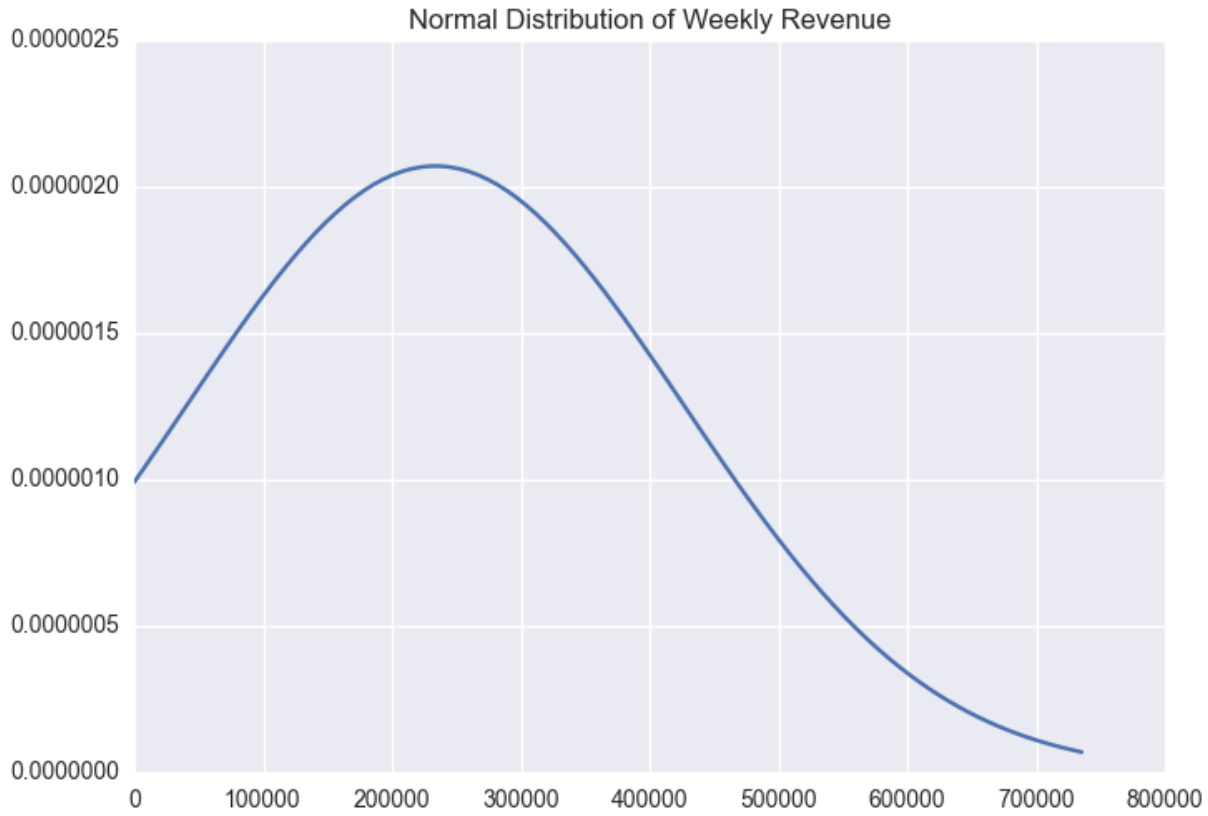
Axes

x: year and month
y: variance



As can be seen variance among month differ significantly.

normal_distribution_weekly_revenue()	
Data	
Tables	**Columns**
Conversions	'timestamp', 'price'
Properties	
dropped 'Nan' and 'None' values from 'timestamp' in Conversions	
Actions	
grouping Conversions on 'year' and 'week', performing sum() operation on 'price'	
counting normal distribution of the data	
Axes	
x: revenue	
y: value	



As above plot indicates Normal Distribution of Weekly Revenue reaches a peak around 2300000.