Market model of a city

Frist printed in January 2013
This book printed in March 2013

BY

V Raghunath Kumar R

About the Author

Raghunath Kumar has finished his masters from IIT MADRAS in mid of 2007 and since then working with TVS Motor Company with Advanced Engineering Group and worked on various technologies. He applied for several patents and presented his work at few international conferences on technical topics during his tenure at TVS Motor Company. His projects include development of **ABS controller, BMS controller, USB stack, DAQ systems** and several other electronic controllers and control & software modules. In addition to technical challenges, his interests include **amateur radio** and **patent rights & law**. In the late 2011, he also released a report titled BUSINESS PROPOSAL TO TVS FOR DESIGN AND MANUFACTURE OF PREMIUM BICYCLES.

The Author can be reached at raghu.elec@gmail.com or can be contacted over phone at +91 74110 RAGHU (+91 74110 72448).

Copyright Information

Preface

Present report is a result of attempt done to model the market behaviour with an influence from agent based modelling methods. The model considers population number for ease of computation, and this may be scaled up/down to suit actual population numbers. Present model is largely deterministic and not yet a completely stochastic model. Randomness in the model includes the variation of some of parameters within known boundaries during simulation run. Each agent in the model is designed to have properties, based on which the effect of each event in the market to shift model parameters randomly within a closed interval is decided. Opinions on all products within the model and need & afford parameters for each agent are dynamically updated with each tick of event based on timeline. Some of real-life characteristics of market such as market sentiments/market drive and aging factor for all opinions are also considered in the model. Overall, results are presented in a single plot window, to comprehend the output of simulation.

About the model

This is a conceptual statistical model with agent based approach for the two wheeler market for an Indian city. This is created with an aim to provide quantitative perspective to the market events and business plans.

Plot Window:

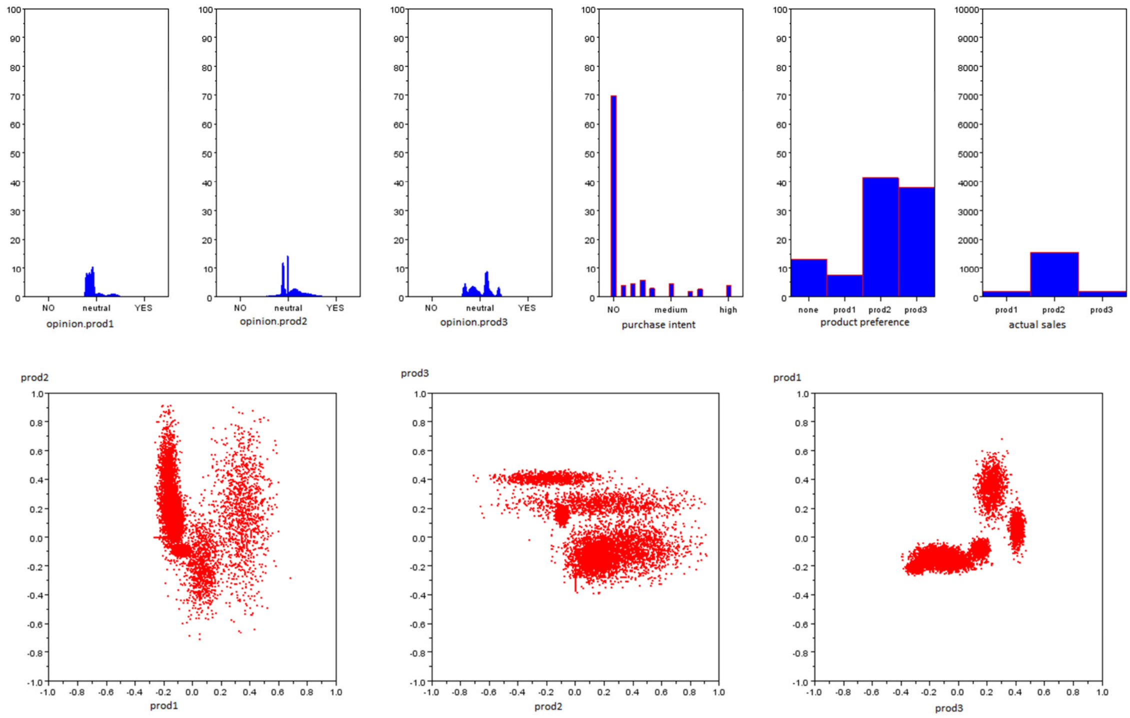


Figure 1: a screenshot of the plot window with all parameters of the sample model simulation at a random tick

- Row 1, plots 1 to 3 :
  - opinion histograms as percentage of total population
- Row 1, plot 4:
  - histogram of purchase intent as percentage of total population
- Row1, plot 5:
  - histogram of preference among all products of simulation as percentage of total population

- Row1, plot 6:
    - Histogram of actual sales estimated using the simulation, with a scale factor
- Row2, plot 1:
    - Scatter plot of opinions between Product3 on Y axis and Product1 on X axis
- Row2, plot 2:
    - Scatter plot of opinions between Product2 on Y axis and Product3 on X axis
- Row2, plot 3:
    - Scatter plot of opinions between Product1 on Y axis and Product2 on X axis
- The number on top of at centre of plot image shows the tick count.
    - Each tick is a time quantum for which simulation is run.
    - In the present simulation, 1 tick is assumed as 1 month.

Aim: to simulate the effects of several events such as product related events, social events, and economic events that effect vehicle sales directly or indirectly, to create an awareness about the market movement and to help devise counter actions for the market situations.

Benefits: Every business plan can be validated with the simulated model for the market reactions, and can obtain quantitative estimate of market movements and hence go ahead with most optimum plan.

Difficulties: Programmer need to quantify the effect parameters of each event, partly based on previous events, partly based on intuition. Several past market events need to be quantized and fed to the model to include effect of past events on the market and arrive at present scenario.

Risks: Every new event provided to the model also needs to be quantified in terms of market reaction. This quantification is left to the person giving this input. Model cannot predict how intense every new event is going to be, and so only the inputs are key factors for market reaction representation. However, there is always benchmark event which would have happened in the past, and the User can always use these numbers as reference to proceed with.

Procedure: A known population set is created and the **demographic** data is provided to the model. Each person represented in the population is referred as agent in the simulation in this report. Population is distributed based on details such as age, gender, affordability range, need range, etc. Each demographic property has certain groups and each group is identified with unique number. Each element of the population has demographic property and other characteristics. In addition, some of the relevant products in the market are considered.

The model at the present state considers three products Product1, Product2, and Product3. Choice of these three is based on following factors. Product1 represents the high performance youth bike from Manufacturer-X, and includes Fiero and all Product1 series. Product2 represents economy bike from Manufacturer-X. Product3 represents high performance youth bike from Bajaj. Focus of present simulation is on Product1. So an economy bike and a competitor performance bike were chosen as co-products. Product2 absorbs all events related to economy bikes and is a generic economy bike as per the scope of present simulation. This scope can be increases by adding additional economy bikes from competitors and then adding separate events related to separate economy bikes to the simulation.

There is also additional detail added to agents in the simulation called opinion about each product. Opinion of agent in the simulation is a rational number in the closed interval [–Xmax, +Xmax]. Xmax value can be set in the simulation window. Precision of the opinion parameter is the simulation environment dependent, and in the present SCILAB or MATLAB environment, it is up to 15 decimal places (for example: - 0.141592653589793). An opinion of 0 represents neutral opinion, positive number represents positive opinion and vice versa. An agent will have opinion about all the products that we consider in the simulation. Agent's opinions are different for different products. Agent's final action of purchase is towards purchase of that product for which the opinion is highest. An example table is given below.

Table1: set of opinions for agents 1 to 10 in a test simulation run

Agent Number	Opinion.Product1	Opinion.Product2	Opinion.Product3	Preference
1	-0.042253829	-0.06493514	0.044950263	Product3
2	-0.056349954	-0.081446831	0.042824051	Product3
3	-0.035071484	-0.104984093	0.052578991	Product3
4	-0.040573654	0.094297131	-0.047347034	Product2
5	0.314829368	0.031457536	0.130739965	Product1
6	-0.11514337	0.772365147	-0.079164259	Product2
7	-0.034000948	-0.073733032	0.059509041	Product3
8	-0.034689908	-0.059987961	0.050803701	Product3
9	-0.154153431	0	-0.111150329	Product2
10	-0.037579274	-0.069821776	0.058052498	Product3

In the table1, the product with highest opinion among all for each agent gains priority for the agent during purchase. When the **need** arises, when afford condition is satisfied, agent **intends** to proceeds with purchase **action**.

Opinion represents the cumulative experience index with the product. This value for a particular product for a particular agent depends on several factors such as agent's demographics and recent market events. Recent market events include product launches with improvements, news about products, economic activities in society, inflation, fuel price changes, city transportation infrastructure, change in public transportation fares, and issues with public transportation. Other factors that affect the opinion about any product are agent's own age group, agent's own gender, agent's social status, etc. there is also an aging factor for opinion introduced in the model, which brings the opinion to close to neutral with passing time. This can be justified by saying that any product without any news or updates or refreshes will lose its public image (positive or negative) and will create neutral image. This may be similar to saying that the motorcycle titled Victor, which has mostly positive opinion some time ago, has mostly neutral opinion across sections of the population mainly because of no news about the product.

Each type of event changes the opinion for individual product in the market differently. For example, launch of upgraded vehicle with increased engine capacity will generally have increased opinion about that vehicle, and this may generally not reduce opinion for other vehicle. However, well circulated news about award for a vehicle will improve opinion for the bike and reduce opinion for other bike.

Need column of the agent's properties represents the necessity for the agent to have personal transportation. This column assumes values from a fixed set of numbers, {41, 42, 43, 44} to represent segregation of population among four different categories of needs, starting from **no need** and ending at **compulsory (**including **optional, required)**. Initial values for need column of all agents are assigned based on age group, working population.

Afford column of agent's properties represents the affordability of the agent to purchase the vehicle of agent's necessity. This column assumes values from a fixed set of numbers, {51, 52, 53, 54, 55} to represent segregation of population among five different categories of affordability, starting from **can't afford** and ending at **petty thing(**including **with compromise, with ease, as an option)**. Initial values for this column are assigned based on age group, and educational qualification.

Intent column of agent's properties represents the agent's intention to purchase the vehicle. This intention does not directly convert in to market sales. Market drive value will be considered and intention above this threshold only converts to sales. Intention value is derived as the product of need and afford. This value is normalised to ease computation at later stages during threshold comparisons.

Just because need and afford are only integer values, intention does not assume fixed set of values. Any market event which alters the need and afford is modelled in such a way that a temporary columns hold the altered values of need and afford which are real fractions, offset from their original values based on the event. Intent is calculated from these values and the need and afford values are updated to the nearest integer. This action is equivalent to rearranging the groups for the population so that the market alterations are included in the model.

Market events are modelled as several types of events, each dependant on certain properties of agents and each affecting certain type of population more than the others. An example market event as presented below is increase of fuel prices. This effect is expected to be in two modes.

Affordability needs to be altered and opinions are to be altered as well. Affordability variation will be such that those in afford-51 will not change. Those in afford-52 will tend to have largest negative slide and those in afford-53 will have smaller slide towards afford-52. Afford-54 will have even smaller change in affordability and afford-55 will have least change or no change.

Also, with increase in fuel prices, opinions are expected to change based on age groups. Age-61 will have least or no impact as they are bothered about price at all. Age-62 will have minimal impact as most of them are just teenagers and majority won't spend their own money in maintenance. Age-63 constitutes majority of people who have just started to earn. Price hike impact would be considerable on this group, since this group constitutes all small wage earners and large wage earners. A mixed change, but slightly small can be expected from this group. Age-64 contains people, who are settled with families and are concerned about savings, whichever is the income group. But also, they are at their highest earning capability. However, due to more cautious nature of people in this group, they tend to react negatively to the price change at the overall level. Age-65 and age-66 have their travel requirements reduced and don't tend to get influenced by price change.

In terms of modelling this behaviour, opinions for mileage bikes are increased and opinions for power bikes are reduced for age groups which are estimated to get affected by price change. Magnitude of change is decided based on the affect enormity. Opinions for mileage bikes and opinion for power bikes are unchanged for remaining groups. As per the model, 0 value in standard deviation means that opinion is unchanged.

Example data input to the model for the presented condition is as below.

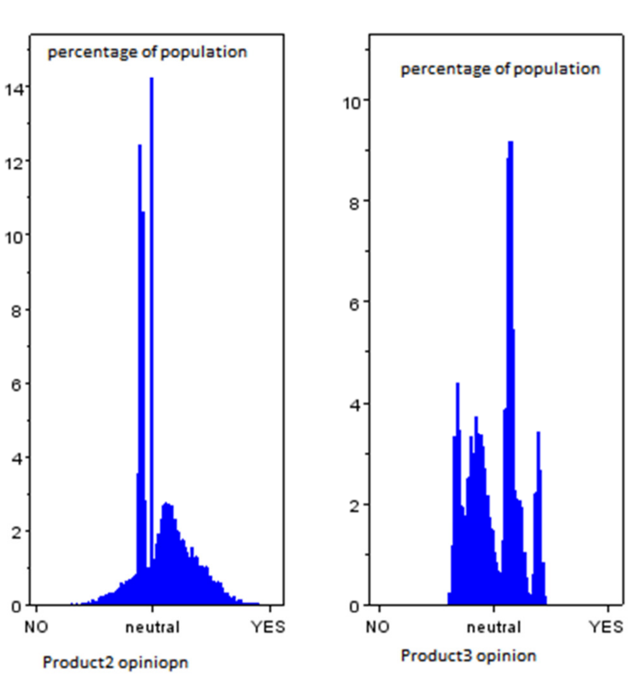
```
Product1.age1.mean=    [  0.000   0.050 0.000  -.040   -.030   0.000 ];
Product1.age1.var=     [  0.000   0.030 0.100  0.020   0.020   0.000 ];

Product2.age1.mean=    [  0.000   0.000        0.010   0.100   0.000   0.002 ];
Product2.age1.var=     [  0.000   0.000        0.100   0.010   0.000   0.000 ];

afford.shift1.mean=    [  0.000  -.500  -.400  -.200  -.100  0.000 ];
afford.shift1.var=     [  0.000  1.000  1.000  1.000  0.500  0.000 ];
```

To discuss more about the model and its architecture, it is in the lines of agent based modelling. The model created does not include neighbourhood influences yet. This will soon be incorporated in to the model. All the agents in the model have demographic properties, and are assigned in to different categories. The way each agent reacts to the market conditions depends on the category the agent belongs to. Category switching is also possible as per the model based on conditions such that affordability can increase or reduce; need for transport may increase or reduce.

The simulation plots the results on a plot window and also optionally generates an animated GIF for viewing the plots later.



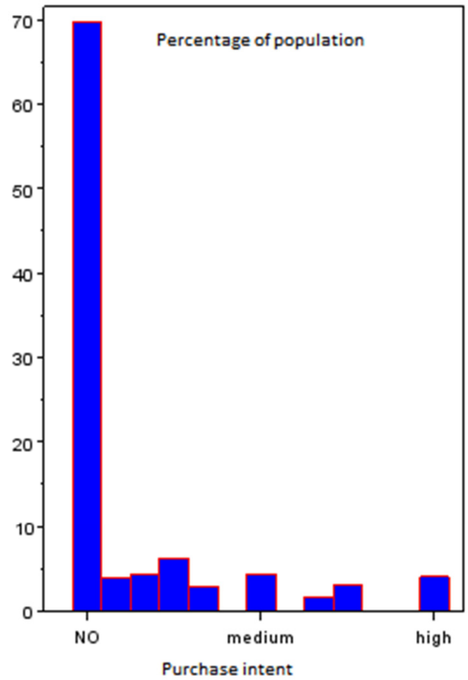


Figure2a, 2b: histograms of opinions

Figure 3: Purchase intent

Opinion is a rational fraction in the closed range [-1, 1]. The histogram as shown in figure 2 represents the population density with the given opinion. For plotting convenience, opinions are discretised to create opinion ranges for x-axis. Y axis plots the percentage of total population with that opinion range. Any opinion on the right of neutral (0 on numeral scale) towards YES (1 on numeral scale) represents a positive opinion on that product, and vice versa.

At the beginning of the simulation, opinions for all the products start with 0 which is equivalent to a single vertical line on the histogram plot at neutral value (0 on numeral scale). With each tick, an event simulation may be done and opinions may get updated. By observing the direction of movements of majority of opinions, we can estimate market response for that event.

We notice the clusters of opinions in the histogram, which are representation of opinions of groups. However, every agent in a group does not carry same opinion about the product. The opinion span differs among groups.

Purchase intent histogram is plotted using purchase intent data of all agents in the simulation. Histogram plots the data as percentage of total population. As mentioned earlier, purchase intent is generated from need and affordability. Purchase intent ranges from 0 to a max of 4 as per the current simulation model and need, affordability matrices. Purchase intent value is updated with every tick of simulation.

Intent is presented in histogram to easily represent changes in intent to purchase. Although intent is presented as rational fraction between min and max values, to represent in a histogram, equal intervals of the value range is considered.

Intent changes with change in need and affordability. Intent, together with positive opinion on any of the product and market drive will generate sales numbers.

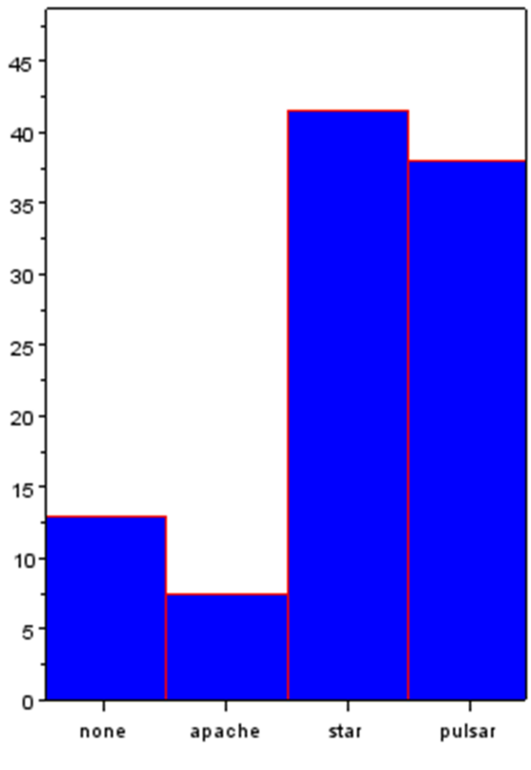


Figure 4: Preference

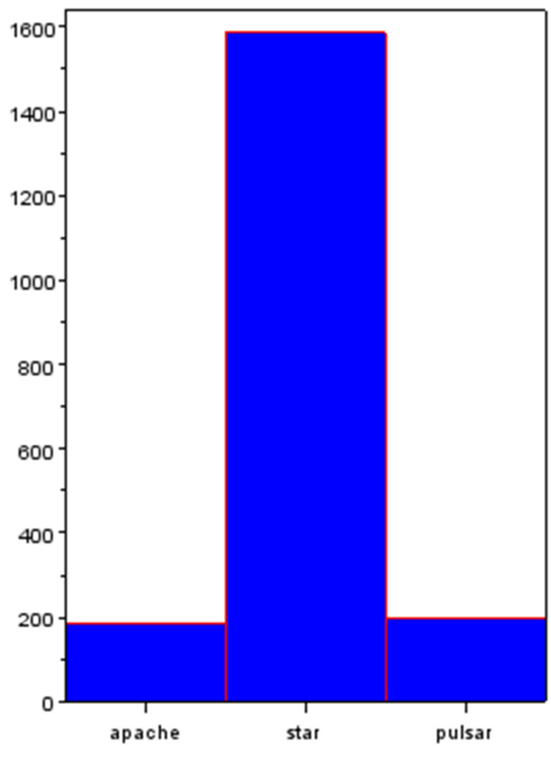


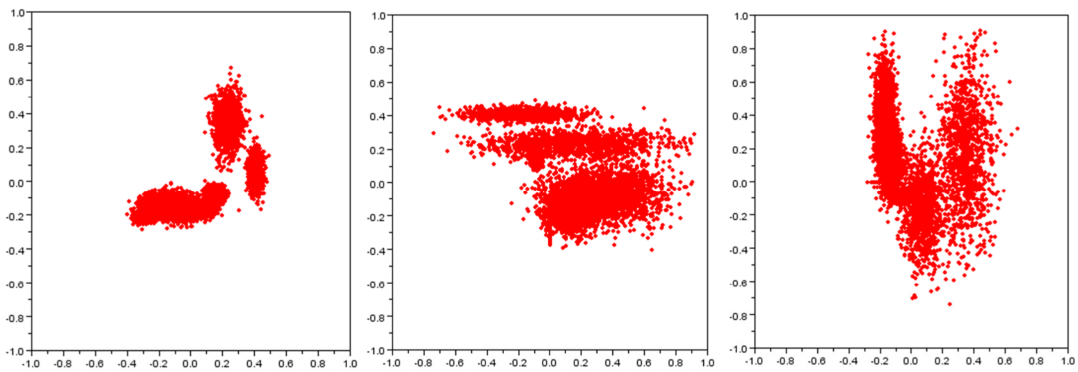
Figure 5: Sales histogram

Preference histogram is plotted using opinion data of all agents. P Histogram plots the data as percentage of total population. Preference data is obtained as highest preferred vehicle model for each agent. Any agent with zero or negative value as highest opinion is considered as preferring none.

As the opinion values get updated with each tick, preference values also get updated. Preference for a product directly does not translate to sales numbers. Affordability and intention to purchase also play a role.

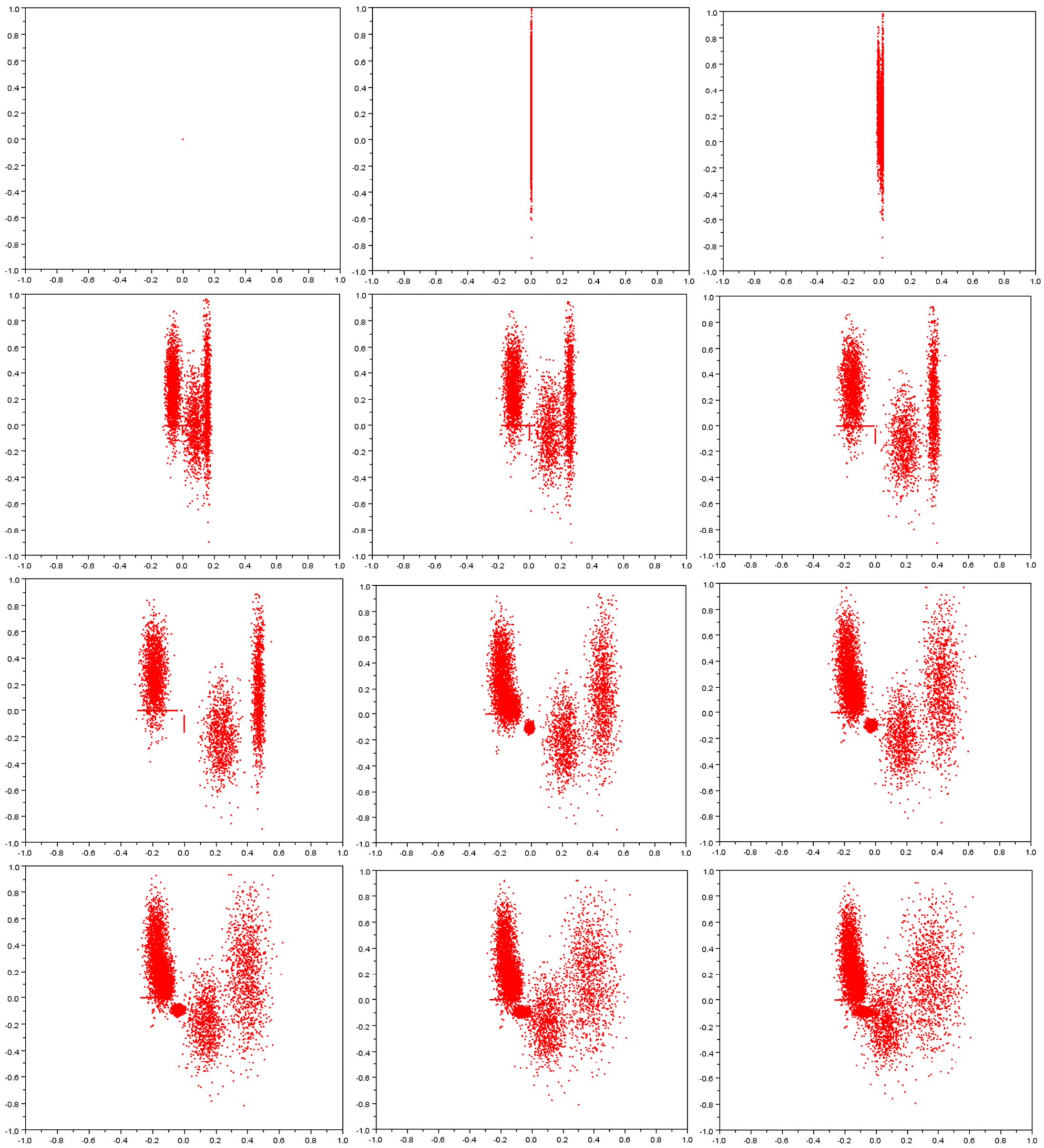
Sales histogram plots actual sales as estimated by the model. Histogram plots the data as actual numbers out of total population of 10^4 in the simulation. Sales estimated are derived from intent to purchase and preference values. For Any agent, if the intent is higher than market drive value considered, and if preference is towards any product, that agent proceeds with purchase of the preferred vehicle.

Market drive value is a dynamically updated real fraction based on market conditions, economy, and general market sentiments towards purchase. Market drive value finally decides the sales numbers. Histogram represents the sales numbers for each vehicle.



Figures[6a 6b 6c] represent opinions of product1,2,3 plotted againest each other as 1-2, 2-3, 3-1 respectively

Figures 6a to 6c represent scatter plots of the opinions for each vehicle at some random tick of the simulation.



Figures 7a to 7l represented as [a b c; d e f; g h I;j k l]

The figures 7a to 7l represent the time line variation of opinions of people between Product1 and Product2 vehicles. It may be very easily noticed that agents form clusters in terms of opinions they have. These screens appear one after another to form an animated sequence and give us visual representation of opinion alteration. First quadrant as per Cartesian coordinate system represents opinions of users who have positive opinion about both Product2 and Product1 and so on.

Analysis and explanation for model parameter selection

This section focuses on analysing the parameters chosen for the simulation for their relevancy to actual conditions in the physical system.

Simulation **population** is considered to be 10^4 which is a reasonably large number to distribute all agents among several groups and demographics. However, to scale numbers to actual population of nearly $8.5*10^6$ we need to include a multiplication factor for final results. Of all the numbers presented at the plot window, only sales numbers are presented as actual numbers, which depend on the population considered in the simulation. Implementing the scale factor for this would help us get the simulated sales number for actual population of the city.

Agents' demographics details considered are age group, gender group, need group, affordability group. Age group and gender group are the key groups which are constructed from census data for Bangalore urban from census 2011.

Age is divided in to 6 groups of identification {61 62 63 64 65 66}, and represents non-overlapping age groups. Age grouping 61 refers to partially closed age interval **(0 14]** to include all kids who do not have any intention of purchase, or who may not have any strong need for personal transport. These are specifically kids up to high-school education, or below.

Age group 62 refers to closed age interval **[15 19]** to include all teenage groups who may have need for personal transport but mostly cannot afford to purchase the vehicle on their own. Majority in this age group depends on parents for purchase, while minority are self-employed with low educational level.

Age group 63 refers to closed age interval **[20 24]** to include all personal who are possibly enthusiastic about motorcycles, and may afford to purchase. Most people in this age group may have just started earning. Some are just out of college with high pay and some have settled in jobs with medium pay. Overall, purchase probability would be high for this group.

Age group 64 refers to closed age interval **[25 34]** to include all married and unmarried people who are a mixed bag of income ranges. This age group is the most diverse and can be uncertain. Those with high educational background will be able to afford the vehicle of their choice, while those with low educational qualifications may belong to medium income group and may be oriented towards saving money and economy products.

Age group 65 refers to closed age interval **[35 49]** to represent all the elders who are still working and may need personal transportation. This group also has several large earners, but only people with high educational qualification are expected to be high wage earners. However, people with lower educational qualifications also would have settled well with their families but are more cautious towards savings. Hence, this age group has most people with orientation towards economy, and may secretly dislike youth bikes and high performance bikes.

Age group 66 represents open age interval starting from **50** to represent all elders close to their retirement time and have settled in a place and may not intend to travel far for daily needs. People in this group may dislike youth bikes for their risky rides as perceived by them. Although people in this age group may not have general preference towards any vehicle, they prefer going with humble rides and are partly sentimental about their earlier experiences.

To consider six age groups to identify all clusters of total population based on age may be a reasonable choice considering the following facts. Most school goers are below age 15 and will only admire, and get inspired. It may be argued that majority of them will have similar influences and similar life styles. Clustering teenagers into one group may be attributed to their thought process on risking things, not caring for maintenance money, and peer pressures about how they maintain their status among the group. Adults between 20 and 24 may be grouped together since they all have reached earning age, except a few who are pursuing their higher education. This age range is below the age for majority of current generation youth to get married and start experiencing monetary challenges. Intention to group adults between 25 and 34 is to include all diverse population with uncertain thought process in to one. Those with high educational qualifications are high earners, and vice versa. This group includes people from various marital states, and various monetary needs. Extreme diverse nature of this group can be used to model opinion range with very wide standard deviation of the opinions. People between 35 and 49 belong to those who settle in their careers whichever it is, and they are mostly not risk takers. Although they are not able to accept that they have crossed the age to risk, some of them may not be able to accept. Also, peer influences are very less with this group, and personal life and money issues takes priority over social status, and building image among peers. This is the age group that mostly focuses on their kids' needs. Those ever 50 years may all belong to elders group who start planning for retirement. Most of these are not certainly risk takers and not with aggressive attitude towards life and their purchases. General sentiment is towards saving money for later use. Most of the requirements may be compromised not because of lack of money or not because of need, but because of the effort required.

Every person in the single age group will not have single opinion about any product in market. This is taken care by spreading the opinion around a mean value with a standard deviation value provided for the normal distribution. Mean value for the opinion is considered based on general tendency for the age group towards the product. Standard deviation value is based on the variety of opinions and the reach of opinions in the age group. Figure 8 represents an example dataset of 10,000 random values whose histogram is plotted with 100 bins. We can notice that the peak of curve is towards 0 and the majority of spread is from -0.3 to +0.3. This is -3σ to $+3\sigma$, which covers typically 99.73% of total population.

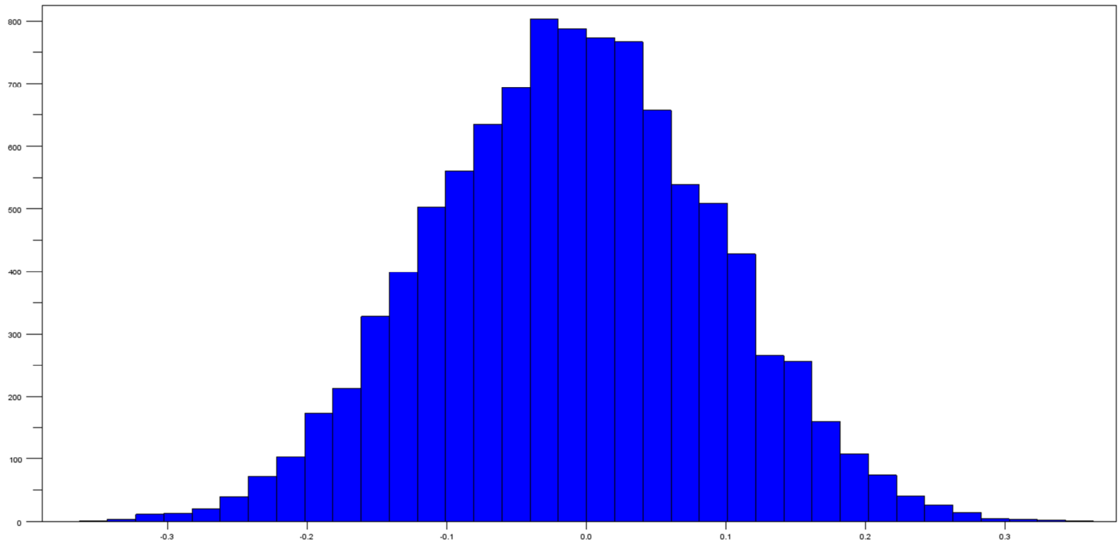


Figure 8: histogram of normal distribution of random numbers with mean 0 and standard deviation (σ) 0.1

Gender is represented as {21 22} to represent males and females respectively.

Affordability is represented with 5 groups of identification {51, 52, 53, 54, 55} while each group represents different affordability condition. Affordability group 51 represents the set of people who are completely **unable to afford** the purchase of vehicles with their then condition. These may include all students, some teenagers, and majority of adults with no educational qualification.

Affordability group 52 represents set of people who can afford purchase **with compromise**. This set includes teenagers with wealthy parents, youngsters with medium educational qualifications, and adults with low educational qualifications. The kind of compromises includes forgoing another important purchase for a vehicle. Purchasing the vehicle will be a short to long term plan and they may also have to undergo some financial compromises while maintaining the vehicle. These may not be able to go for another purchase in a very long time and prefer to maintain the vehicle for long time.

Affordability group 53 represents set of people who can afford purchase **with ease**. This group includes people with different needs but majority of needs covered with money inflow. These set of people are in a position to afford a purchase at the expense of no savings, but not at the expense of another purchase. These are the people from middle income groups and majority aspire for savings as a general sentiment. However, some purchases may also be made to show off to their peers to establish superiority. Overall, this group may be oriented towards economy and performance vehicles equally.

Affordability group 54 represents set of people who can afford a purchase as **an optional mode of transport**. These are set of people who need not stop savings to go ahead with purchase. Loan requirement may also be close to zero for a purchase from this category. Majority of people from this group do not put logic for purchase and they go with admiration towards the product and peer influence. People from this group don't worry about mileage and maintenance expenses. They only worry about comfort and style.

Affordability group 55 represents a set of people who can afford a purchase just **at their will**. These people are high earners and have high money at their expense. People from this group worry nothing about mileage, fuel prices, and maintenance charges. They only go with luxury ride comfort, and attitude of the ride. These people purchase the vehicles as a backup mode of transport and do not have much use in their daily lives. Market fluctuations do not have any effect on their purchases and their purchase is driven by intuition, and influence. These impulsive buyers take their influences from peer circles, mass media for the rich and social media for the youth.

Affordability has five groups, with each targeting different income levels. Affordability is derived from Census 2011 data for Karnataka Urban Educational level vs. age. Data was provided for educational range of age groups. Bangalore is estimated to be very similar to Karnataka Urban, since the overall lifestyle and city facilities are very close to the urban zone of overall Karnataka. Hence, it is assumed that same percentage distributions apply to Bangalore as well.

From this data, required age clusters are clubbed and percentage of population for each age group and each educational qualification was obtained. It was assumed that educational qualification and age group combination will translate to affordability. For example, those belonging to 25-34 with matriculation as their educational qualification may only belong to low income group people and may be struggling to survive with family and its additional expenses. But with that educational qualification, these are the people who have lived and settled in whatever they do, and hence would be earning reasonably to cover their needs. So, they could afford the purchase of vehicle with ease by compromising on their savings, although they may not have to give up other critical purchases.

Similar thought process was used to derive values in table 3 from table 2. Table 4 data was the aggregation of all data from table 2 and table 4 to cluster all affordability based on age group into one.

Table 2: Educational level as percentage of population vs. age

%	no edu	Literate without educational level	Below primary	Primary	Middle	Matric secondary	Higher secondary	Non-technical diploma	Technical diploma	Graduate and above
0 to 14	51.0382	0.27547	26.3366	17.9028	4.44694	0	0	0	0	0
15-19	17.4285	0.92271	7.2389	18.3397	22.3753	25.621	7.40164	0.09597	0.57621	0
20-24	22.9466	1.20568	7.03056	15.5912	10.9183	18.693	13.004	0.35956	2.10866	8.14238
25-34	34.598	1.35607	8.37967	14.8491	7.28273	14.681	6.82947	0.28267	1.70419	10.0372
35-50	45.0904	1.47603	10.1069	15.4013	4.86935	11.4058	3.4416	0.13044	0.88359	7.19466
50+	58.6689	1.56139	10.7893	12.6453	3.1952	7.07149	1.44445	0.12985	0.7091	3.78497

Table 3: affordability index for population vs. age group

	no edu	Literate without educational level	Below primary	Primary	Middle	Matric secondary	Higher secondary	Non-technical diploma	Technical diploma	Graduate and above
0 to 14	1	1	1	1	1	1	1	1	1	1
15-19	1	1	1	1	1	2	2	2	3	3
20-24	1	1	1	1	2	2	2	3	3	4
25-34	2	2	2	3	3	3	3	4	4	4
35-50	1	2	2	2	2	2	3	3	4	5
50+	1	1	2	2	2	3	3	4	4	5

Table 4: constructed affordability range segregation of population

	1	2	3	4	5		
0 to 14	100	0	0	0	0	1	cant afford
15-19	66.3	33.1	0.6	0	0	2	with compromise
20-24	46.7	42.6	2.5	8.2	0	3	with ease
25-34	0	44.3	43.7	12	0	4	as an optional mode
35-50	45.1	43.2	3.6	0.9	7.2	5	at will
50+	60.2	26.6	8.5	0.9	3.8		
			affordability matrix				

Table 5: Raw data from Census 2011 records for Karnataka Urban educational details

S No.	Age groups	Total population	Literate without educational level	Below primary	Primary	Middle	Matric secondary	Higher secondary	Non-technical diploma	Technical diploma	Graduate and above	Unclassified
1	0-6	7182100	0	0	0	0	0	0	0	0	0	0
2	7	1023407	6157	709223	0	0	0	0	0	0	0	0
3	8	1288401	5634	1065300	0	0	0	0	0	0	0	0
4	9	1130117	3771	965241	50680	0	0	0	0	0	0	0
5	10	1462506	6433	959046	300552	0	0	0	0	0	0	0
6	11	1016642	3586	338923	590896	0	0	0	0	0	0	1
7	12	1426809	7788	216303	933922	58486	0	0	0	0	0	1
8	13	1089999	5279	88529	687824	187742	0	0	0	0	0	0
9	14	1225620	7756	94001	451956	502885	0	0	0	0	0	0
10	15	1103115	8449	80251	241037	451873	136149	0	0	0	0	0
11	16	1186668	10448	90245	225824	302358	351327	0	0	0	0	1
12	17	756205	5832	46190	124833	129195	361588	0	0	0	0	0
13	18	1565363	17510	122265	274959	215187	352783	222223	2653	15662	0	1
14	19	770321	7418	50623	120331	105551	176993	176109	2512	15348	0	1
15	20-24	4886225	58912	343529	761822	533493	913384	635406	17569	103034	397855	11
16	25-29	4578219	60712	364865	684749	368501	714440	355468	14064	80948	483173	6
17	30-34	3821205	53190	338979	562491	243206	518677	218168	9679	62194	359891	7
18	35-39	3918079	57312	385721	609008	206727	455134	155899	5668	40357	292995	3
19	40-44	3058456	45939	307372	468568	143557	350790	100450	4014	25600	223813	2
20	45-49	2713842	39782	286299	414867	121574	299340	77155	2958	19666	180382	2
21	50-54	2109655	31736	217727	291781	82188	207205	47828	3056	17677	121126	0
22	55-59	1423742	21089	163296	200117	51734	113984	23879	2550	14127	62865	2
23	60-64	1498909	24074	154697	180030	43855	89706	15934	1934	10547	46038	0
24	65-69	953187	14863	108093	116289	27396	55414	9203	1111	5560	26845	0
25	70-74	799497	13058	85308	86663	18328	36244	6128	623	3310	16313	0
26	75-79	353230	5853	42660	42094	9602	17611	3101	308	1400	7602	0
27	80+	457199	7921	47714	43491	9586	16945	3639	281	1238	6695	0

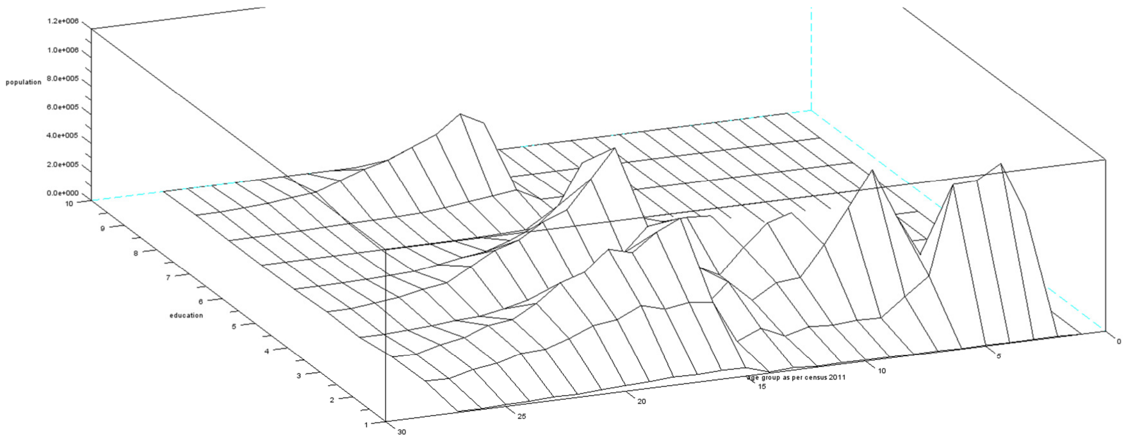
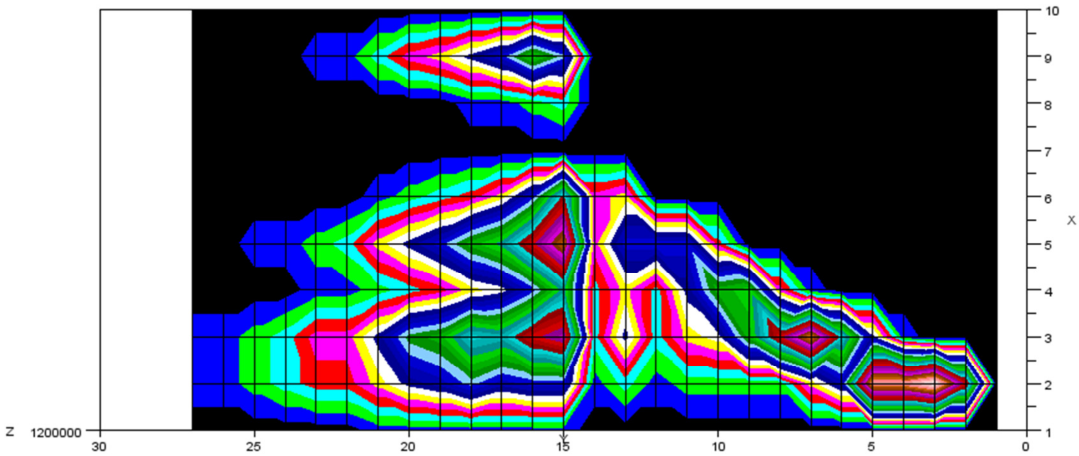


Figure9: data from table 5 plotted in mesh plot



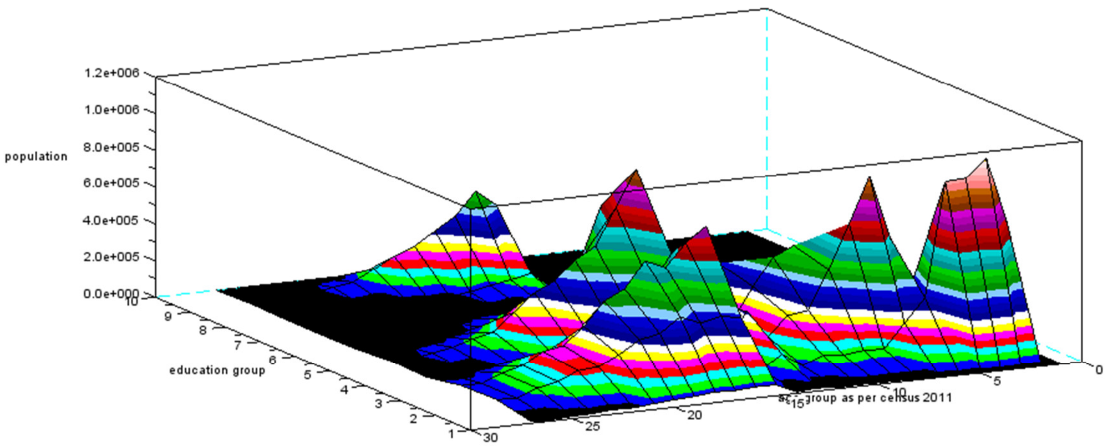


Figure10: Same data as in table 5 plotted (X-education Y-age group as per census 2011, Z-population from KAR urban)

As noticed in the mesh plot and surface plots, there is a clear trend of age group and educational qualification. We may as well notice the population related to each sub section of the distribution from the population.

Need is represented with four groups of identification {41 42 43 44} while different group represents different needs for personal and dedicated transportation. Need-group 41 represents the set of people who have absolutely **no need** for personal transportation. There are different categories of people under this group. This group includes school children and college students, who get their daily ride arranged by respective organisations. Older population who have reached age beyond retirement and have no need to travel long will also fall under this group. Homemakers and other people who hardly go out might fall in this category. This group also includes people who already have a personal transportation medium.

Need-group 42 represents the set of people who need personal transport as an **alternative**. This group includes people mostly from working class who have the comfort of private transport or public transport but may need personal transport when they need comfort of time. These are the people who need to reach their destinations in the time of day when its most comfortable to travel. So, choice of personal transport or public transport is based on their intuition. Market sentiments, and peer influences may drive the need for personal transportation higher.

Need-group 43 represents the set of people who **require** personal transportation more than those already discussed about. These are set of people who may need to reach their destinations without the luxury of time. Public transport may create delays due to time to wait for ride and longer routes taken by public transport. These are the set of people who may find it easy and comfortable in terms of time and money to travel by personal vehicles. Also, these are the people who may incur loss indirectly in economic terms when they depend on public transport.

Need-group 44 represents the set of people who **require** personal transportation **unconditionally**. These are the people who measure their time with economic sense. These people may need to travel at odd times when public transportation may be scarce, or to odd places where reach of public transport is less. These are also the people who may need to struggle to travel in the public transport, while their condition does not support it. For example, a white collar worker travelling in a BMTC bus from K.R Market to Jayanagar to reach office at 10AM. He may feel exhausted travelling standing in the bus and he really needs to be present himself well at his office.

Incorporating another need segment would only mean redundant segment with repeating behaviour. The four segments considered would sufficiently cover all the possibilities of needs. Even if one of the need range is omitted, is would mean too uncertain group with multiple behaviours. Hence choice of group count is justified.

For the present model, three products in the market are considered, Product1, Product2 and Product3. Selection of products is based on the factor that the product of focus and its competitor along with all other products need to be considered. If the competitor vehicle is not considered in the market portfolio, every event only has two effects, one on the product of interest and other on all other products. This situation could be slightly confusing to provide model inputs and for tuning. Since the rest of vehicles element will have to include effects of several events at the same time, quantitative representation of the same could be very complicated.

Here, Product2 represents all economy bikes, and all other products in the market. All the effects of mopeds related market events, and scooter related market events are ignored as far as the present scope of simulation is considered. To include that effect as well, those products also need to be included and events related to those also need to be included for the simulation run. Additional effort required to add additional vehicles to market portfolio is minimal.

Quantitative model of each event in the market will be a matrix indicating the shift of opinions/need/afford based on the demographics/category of the agent. The result of the shift will be sum of previous value and a random value generally located anywhere within $\pm 4\sigma$ of around the μ value of the quantitative model. Quantitative model specifies different μ and σ for each value of demographics/category as shown in the matrices earlier in this report. This is attributed to the fact that different people from different categories react differently to each event. The shift in opinion/need/afford value is very selective based on their environmental conditions which include demographics. Also, based on the diversity of the group, spread of the shift is decided.

About the simulation

Simulation is discretely run for every tick. A tick is a time quantum considered for the simulation purposes. It depends on the condition of simulation and time resolution of events. For the current sample model, a tick is considered equal to one month.

Every event is programmed to have preparation time, peak event, and trail. Each event is programmed to have its characteristic decay coefficient. Hence, an event's start time, peak occurrence time, and rate of decay are programmed for each event. Higher the decay coefficient for an event, shorter is the effect of the event after the peak. Event's timeline shape is programmed based on the type of event and the earlier acceptance of events.

Some events may be short lived and some events may have longer effect. Figure 11 presents four types of events, plotted with different colours as considered in the sample simulation. All the event effects are normalised to 1 with a peak effect as 1 and the effect of the event is given as input to the model for simulation accordingly. When the event's effect is a fraction less than 1, the effect is also multiplied by the fraction. i.e. when the event's effect for the given time is only 0.6, and the event says that the opinion_Product2 for age group 62 has to shift with a mean 0.05, and standard deviation 0.1, the resultant effect would be a mean of 0.03 and a standard deviation of 0.06. This is included to avoid considering constant effect for each event.

With each tick of the simulation, if there are no market events, it is practical to assume that the opinion about the product goes towards zero whether earlier opinion was positive or negative. It may be attributed to the fact that if the earlier opinion was positive and there is no event that is related to that product, people tend to lost the positive opinion. Likewise, when earlier opinion was negative, people might also tend to forget what was considered negative, with time. This may be because of forgetting what they hate/like about it.

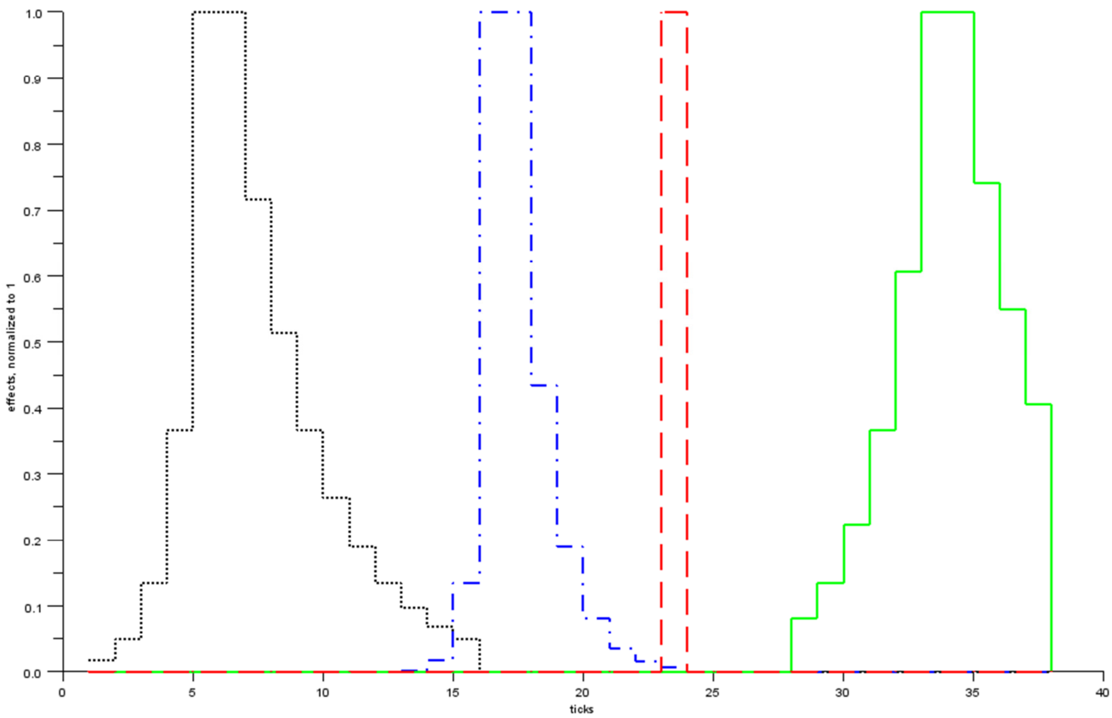


Figure 11: Timeline of events and their effect durations.

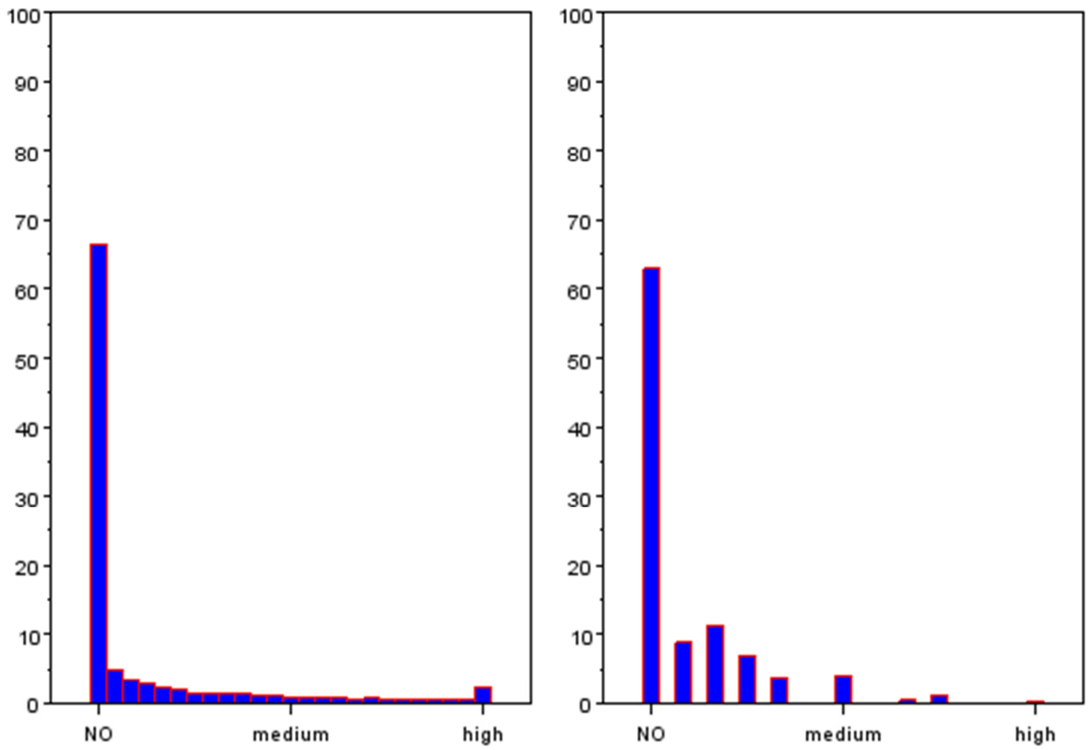
This was captured by introducing **AgingFactor** in the model, which brings the opinions of all agents for all the products close to zero by a predefined percentage. Any market event will have to impose stronger effect in the population in order to overcome the AgingFactor's contribution. This implies that any product which has no market activity will slowly lose goodwill in the customers' mind and may be very much equal to a new product to the market after certain time.

Simulation calculates updated opinions for each tick based on market events. Simulation also updates groups, if needed. Based on NEED and AFFORD values, an INTENT value is calculated for each agent in the simulation. Intent value is directly proportional to product of afford and need.

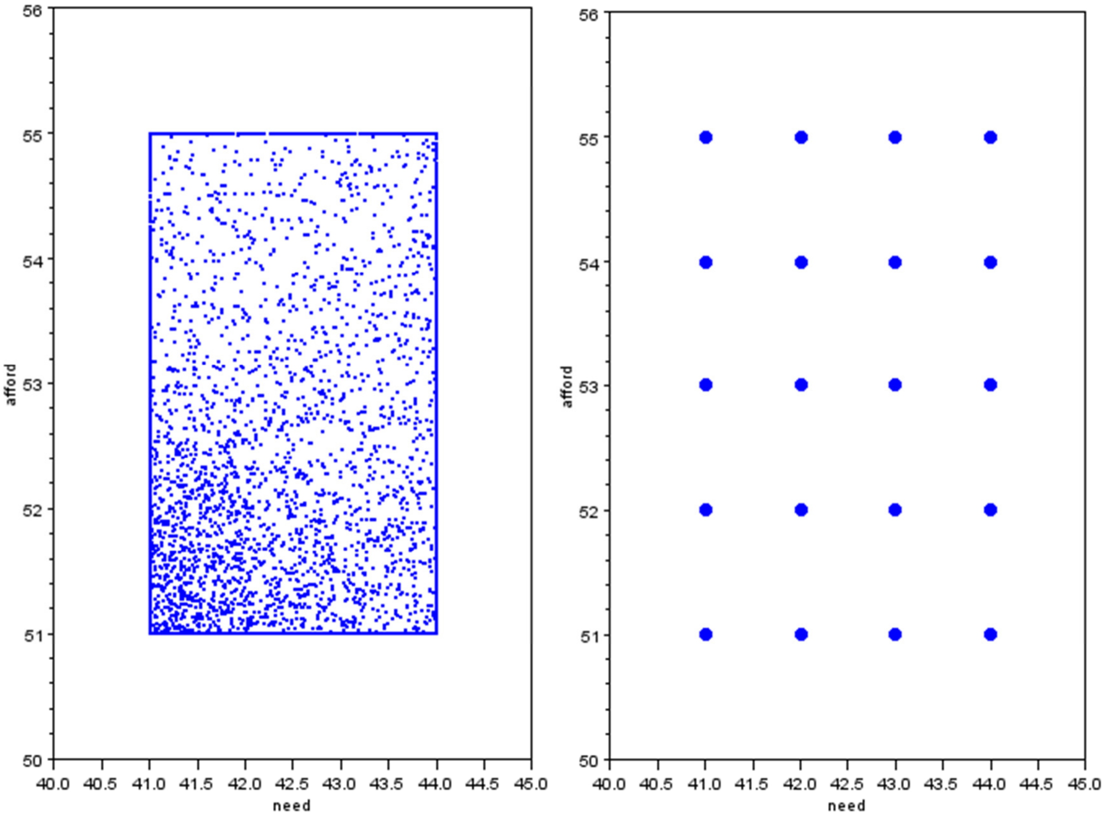
A new parameter called **PurchaseThreshold** is introduced to include all market sentiments for the season. This value will be updated based on time, based on some events' effects to reflect market drive to proceed to purchase from intention. This may be identified with the fact that although people are in need of a vehicle, and they can afford the same, they wait for the right season because of few offers. Or, people might postpone their purchases because of foreseen recession. This parameter is introduced in the simulation to reflect general market sentiments, and is a dynamic variable.

When the agent's intent is higher than the PurchaseThreshold the agent proceeds with purchase of product with highest preference. It was assumed that when an agent's highest opinion for all the products is 0, i.e. the agent has neutral opinion about one and negative opinion about the rest, it was assumed that the agent may not proceed with purchase. Certain market events are programmed to be affecting the population distribution in some segments, such as high employment rate increasing affordability of certain age group people.

Intent for an agent may be assumed to hold only values from a limited set, because of their generation from Need and Afford ranges. In such cases intent will be a discrete matrix and hence altering PurchaseThreshold may only have effect at few values only.



Figures [12a 12b]: Intent distribution continuously and discretely, taken from random ticks of different simulations



Figures [13a 13b]: need vs afford distribution continuously and discretely, taken from random ticks of different simulations

To overcome this drawback, all Need and Afford variations are subjected to shift with a mean and standard deviation of necessary magnitude and before the new values are rounded to their nearest groups, the fractional values are used to calculate the intent of all the agents. This results in continuous intent values as opposed to discrete intent values.

Figures 12a and 12b represent the discrete distribution of intent and continuous distribution of intent.

Figures 13a and 13b represent the distribution of continuous values of need and intent as opposed to discrete distribution of those values.

Tuning the model

The simulation model developed is not tuned yet. In order to tune this model, timeline of all past events need to be collected. In addition, every event needs to be analysed in terms of the type of people it is affecting and the result of such event, and contribution of each group to that result. Once these are quantified in the model EventScale needs to be created with event number, its start tick, peak tick, and decay coefficient. The model's time loop needs to be updated to incorporate the effect of each event and update all the relevant values. To check the validity of tuning, sales data estimated needs to be matched with actual sales generated at the timeline. If these numbers do not match, tuning further needs to be done either by altering the effect of the event or by including any other additional effect of that event on different values of the simulation. This step qualifies to be a verification step.

Once the model is tuned to satisfaction, the any estimated event needs to be created with the proposed effects, based on the learning in tuning the model, and model needs to be run. The model result will be as accurate as the tuning process, and will be totally dependent on the understanding of market event's effect on different sectors of the society. Any missed effect will lead to wrong result.

Flowchart

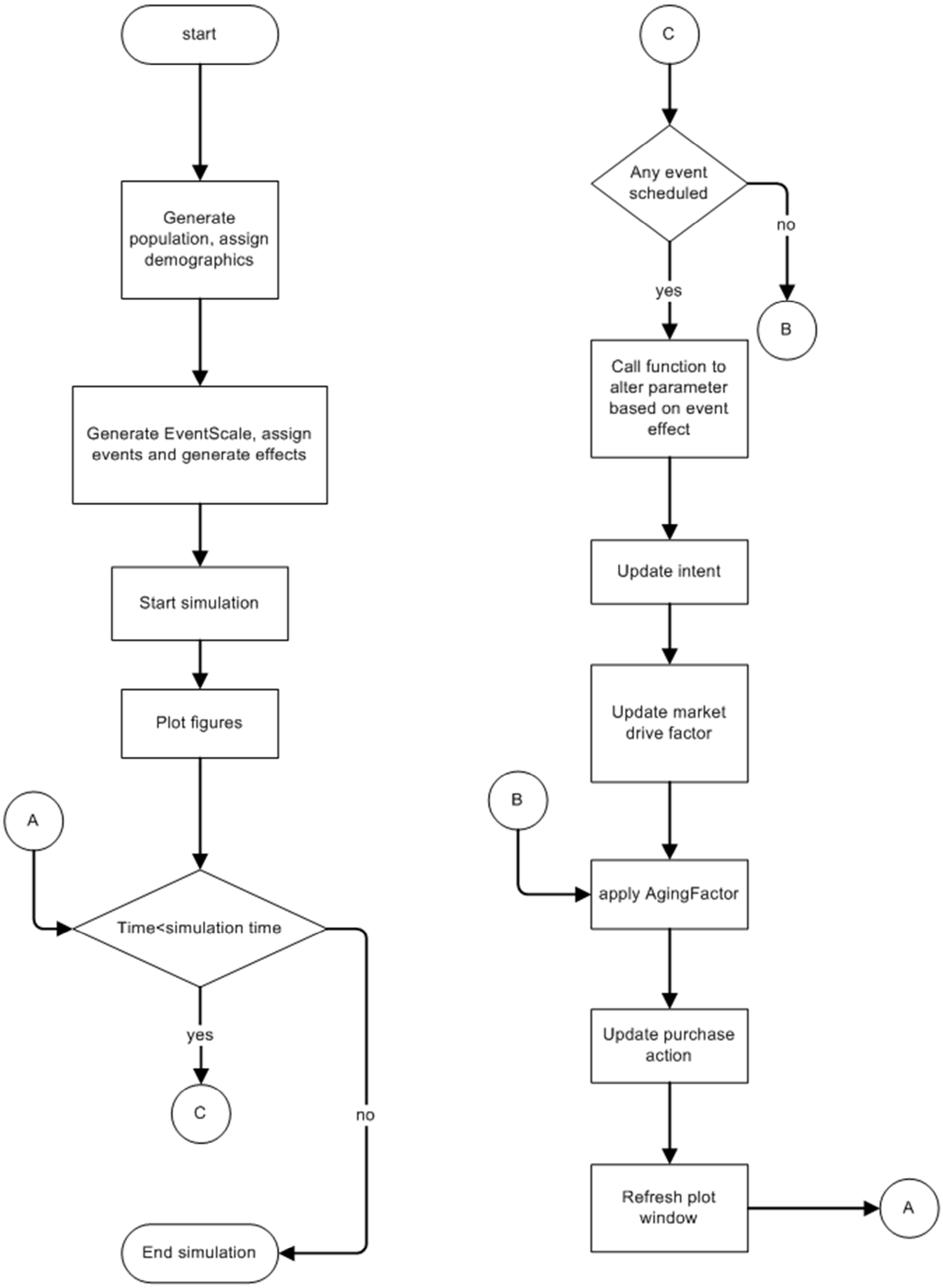


Figure 14: Flowchart of simulation process

Statistical model for estimation of consumer preferences

Venkata Raghunath Kumar Rachabattuni and Samraj Jabez Dhinagar
Advanced Engineering Group, TVS Motor Company, India
{Raghunath.Kumar, SJ.Dhinagar}@tvsmotor.co.in

Keywords: Mathematical Modelling, Consumer Market, MATLAB/SCILAB Simulation.

Abstract: This paper attempts to present a model of consumer market for urban population, created to provide quantitative perspective to the market events and business activities. A population model has been presented with similar demographics as those of a city, and quantitative models for market events are provided to the model. This paper also discusses the method of creating the events and applying them to the model to obtain results. Present paper also discusses the effect of each event on several clusters of population.

1 INTRODUCTION

Predicting how the market perceives a business event such as product release, price increase is always a challenge to policy makers. This becomes even more challenging when the population at focus is diverse in different aspects. It is very important for decision makers to quantitatively understand the effect of a business decisions on the market orientation for the near future, to drive the market in favour of their product. Few models based on agent based methodology have been studied that focus on consumer market. However, sentiments towards purchase are affected by large varieties of market events.

A consumer behavioural model for a urban population has been developed that can accept quantitative model of generic events. Once the model parameters are tuned to match sales outcome and set of events in the past, the model would be useful to estimate market reaction for any event whose effect could be quantified.

2 THE MODEL AND ITS PARAMETERS

The model has been implemented in SCILAB environment, which is a numerical computational tool, very similar to MATLAB. Same model can be ported in MATLAB with little additional effort. All the data required to populate the model is provided as matrices of appropriate values.

This model has been greatly influenced by agent based modelling techniques. The present model alters agents' data based on market event, randomly within the limited interval.

2.1 Population

A population set of large number has been considered and demographic data has been applied to the population to resemble the actual population of the city. Few properties have been given to the population based on the demographic distribution. Care has been taken to distribute population so that agents with similar demographic properties do not cluster together when listed with their serial number. Agents react to market events and alter their opinion about the products in the market. The alteration of opinion is event driven, and will be different for all agents.

Agents' opinion values are considered to be real fractions in closed interval [-1, 1], and can assume any value. Every agent has as many opinions as there are products in the market. An agents' opinion towards a product is 0 if he is unaware of the product. A positive value is indicative of favourable opinion about the product and vice versa. An agent can also have positive opinions for multiple products. That directly translates to real world as a person liking two or more vehicles in the market.

However, when it comes to purchase, agent prefers to purchase the vehicle which has highest positive opinion. An agent with highest opinion as a negative value would mean that the agent dislikes all vehicles in the market. In such cases, model does not consider a purchase event from the corresponding agent.

2.2 Market Events

2.2.1 Timeline

Simulation of the model is run with each tick of time, which is one month for the purposes of the present model.

Model starts with an assumption that few events have already occurred and the model proceeds with further events. To account for this, a step to assign the pre-model values to the population is conducted from the pre-assign matrices. The pre-assign matrices are predominantly set of matrices for each property of the agent to distribute the properties across different property groups by the percentages specified in the matrices.

2.2.2 Events

Designing events is very critical to the model and to run the simulation. Every event constitutes effecting unit and effect values. Effecting unit is essentially the classification of the agent, such as gender.

An event's effect is not uniform on all the agents. For example, release of new vehicle targeted at female population will have an impact on female population, while it does not have so much impact on the male population. Extent of effect of an event on the sub groups of the classification is specified in effect values of the event. Effect values include the range of values for the property change. That is to say that all females in the earlier example will not have same impact, but the impact may range from a lower limit to an upper limit, when attempted to quantify.

2.3 Agents' Groups and properties

2.3.1 Groups

Agents are allocated to different groups based on demographics. These groups include age and gender. Each group has few categories and each agent is assigned a value representing the sub group independently.

Age group has six sub-groups to discretely represent total population clustering them based on emotions, actions, and attitude. Gender group has two sub-groups to represent males and females separately to represent the difference reactions to certain market events.

Overall combination of assigning the above two properties include twelve different reactions possible for each event. Any other additional property would increase the number of possible types of reactions.

2.3.2 Products in the Market

The products in the market are assigned few attributes based on which agents form their opinions. For example, a motorcycle in the market titled p1 may be designated as mileage oriented, with moderate styling for daily commutation with average power. Another motorcycle in the market titled p2 may be designated as a powerful motorcycle with a lot of luxury features and may not return much mileage. Another product in the market, which is a scooter, titled p3 may be the only product marketed for female buyers with reasonably comfortable features.

Any product in the market acknowledges different market events differently. To consider an example, a market event increasing fuel prices may trigger increase in preference towards p1 as compared with p2. But p3 being the lone product of the category, may not have any alteration of preference levels among population.

2.3.3 Opinions

Each agent of the simulation carries as many opinions as there are products in the market model. Every opinion is a real fraction and it is fair to consider that no two agents have same opinion unless the opinions are 0, based on the fact that SCILAB considers real fractions to a very high precision.

Any market event that alters the opinion will specify a range of values for the opinion to change based on the property of the respective agent. Final opinion value for that agent will be separated by initial opinion value by a random value which lies within the range specified in the market event's effect values.

Product of preference of each agent depends on all three opinions and the preferred product is updated for each agent at the simulation runtime.

2.3.4 Intent

Intent of purchase is a combination of affordability and need of an agent. This value is derived as a product of need and affordability values and then normalised. Each of these values is represented as a number indicating the intensity of the feeling. A normalised intent value for each agent is a real fraction between 0 and a max value defined based on source values.

Intent for the agent drives the agent's purchasing behaviour but it may not totally be responsible for the actual sales in the market environment.

2.3.5 Purchase Action

Purchase action is represented as the final outcome of the simulation, which is the action taken by the agents for purchasing a product in the market or not.

A parameter called representing market drive is considered in the simulation to represent the variation of sentiments for all agents over time.

2.3.6 Modelling

Need value is represented as η_p, affordability is represented as α_p, and intent is represented as τ_p, where the value p represents the identification of agent, usually a serial number. Opinion of each agent towards each product in the market is represented as ϕ_p^q, where p represents the identification of the agent, and q represents the identification of the product. It may be understood that the above symbols without subscripts may represent the cumulative property of all agents together.

$$\phi^{pr1} = \{\phi_p^{pr1} | -1 \le \phi \le 1\} \tag{1}$$

$$\eta = \{ \eta^s | 1 \le s \le S_\eta\} \tag{2}$$

$$\alpha = \{ \alpha^s | 1 \le s \le S_\alpha\} \tag{3}$$

In the equations 2 and 3, S_x represents the total number subgroups available for the respective property x of the agents. An event in the simulation is represented as ${}_{m,s}^{n}\Upsilon$, where s represents the attribute of the agent which gets affected, m represents the identification of the agents' attribute which decides the effect of the event on different ranges of the attribute s, and n represents the identification of the event. For example, an event whose effect depends on need group distribution, and affects the opinion on product 1 (pr1) for all agents based on need group distribution, it may be represented as ${}_{\eta,\phi^{pr1}}^{E1}\Upsilon$.

The event ${}_{\eta,\phi^{pr1}}^{E1}\Upsilon$ tends to alter ϕ^{pr1} to $\phi1^{pr1}$ where $\phi1^{pr1}$ is separated from $\phi^{pr1}$by a set of values specific to each agent in the simulation. In most general case, since opinions do not alter drastically and are based on earlier opinions, the final opinion is shifted from earlier value.

$$\phi1_p^{pr1} = \phi_p^{pr1} + \xi_{p,pr1}^{E1} \tag{4}$$

In the equation 4, $\xi_{p,pr1}^{E1}$ represents the variation in opinion for the agent identified with subscript p for the event identified with superscript E1.

The event ${}_{\eta,\phi^{pr1}}^{E1}\Upsilon$ decides the set of values for ξ^{E1} based on each agent's η value. Since all the agents in the η group do not alter opinions by same value, a standard deviation $\left({}_{sg}^{E1}\sigma\right)$ in the value to be shifted is assumed in addition to the mean value $({}_{sg}^{E1}\mu)$ shift which is a result of the sub-group's (sg) characteristics and the event effects.

Every event ${}_{\eta,\phi^{pr1}}^{E1}\Upsilon$ constitutes a set of values for ${}_{pr1,\eta^s}^{E1}\mu$ and ${}_{pr1,\eta^s}^{E1}\sigma$ where superscript s for η represents the sub group identification within the η classification and pr1 represents that the effect is on the opinion of product 1 from the simulation.

$${}_{\eta,\phi^{pr1}}^{E1}\Upsilon = \left\{\{{}_{pr1,\eta^s}^{E1}\mu,\ {}_{pr1,\eta^s}^{E1}\sigma\} \,\middle|\, 1 \le s \le S_\eta\right\} \tag{5}$$

$${}_{\eta,\phi^{pr2}}^{E1}\Upsilon = \left\{\{{}_{pr2,\eta^s}^{E1}\mu,\ {}_{pr2,\eta^s}^{E1}\sigma\} \,\middle|\, 1 \le s \le S_\eta\right\} \tag{6}$$

In the equations 5 and 6, S_η represents the number of subgroups available for the agents' property η. Equations 5 and 6 represent the variation of event's effect on agents' opinions for different products of the market.

Alteration of opinions based on equation 4 is not always a representation of real time behaviour. Sometimes, a complete opinion transformation may occur which specifically depends on event type. In such cases, equation 4 may not be used and alternatively, a new opinion distribution based on agents' property is assigned to all the agents.

The structure defined for events in the model ensures flexibility to define the magnitude of event's effect and also the span of effect. Since any event in the real market is expected to have its effect in several ways, the model considers the most prominent effect of the event for the simulation purposes. This simplification allows the linearization of the event's effect on the simulation population as against the non-linear and highly complex real life behaviour.

When an event of type ${}_{\eta,\phi}^{E1}\Upsilon$ is given as input to the model, the definition to the model includes ${}_{pr,\eta^s}^{E1}\mu$

and ${}_{pr,\eta^s}^{E1}\sigma$ for all the products in the market for all the subgroups of η. This ensures that the effect of the event across complete market is provided as input to the model. The simulation process makes sure that the effects are applied at the pre-set time of the simulation.

Any event from the real life can be modelled using the linearization method as discussed earlier to incorporate its effect on agents' opinion change about all the products.

Few events also consider alteration of η groups, α groups, or some other property groups to replicate few types of events in real market such as infrastructure changes, economics of the market, etc.

$$\tau_p = \frac{\eta_p . \alpha_p}{k} \quad (7)$$

$$k = \langle \eta \rangle \langle \alpha \rangle \quad (8)$$

The symbol ⟨x⟩ from equation 8 represents the mean value of the set with values {x}.

Consider that ε_p represents the preference of agent p. Each agent's product preference may then be updated at the end of each simulation time quantum to reflect the effect of events in the past time step on the market behaviour. The parameter ε_p stores the identification value of the preferred product for the agent p from the model.

$$\varepsilon_p = \begin{cases} y \mid x_y = \max(\phi_p^x) \\ 0, if\ y < 0 \end{cases} \quad (9)$$

Equation 9 represents that the preference for each agent would be for a product for which the agent processes highest opinion or none, if the highest opinion is still negative. This was done to ensure including real life behaviour that a person without a positive impression about any product may not proceed with purchase even when need arises.

Once the model calculates the intent τ_p and updates agents' preference ε_p, purchase action may be calculated. Consider that λ_p represents the final purchase action of the agent p. The parameter λ_p stores the identification of the product that the agent p from the model has purchased.

It is known that purchase intent alone does not result in purchase action. The model also assumes a parameter called purchase threshold which is represented as β. The parameter β is dynamic and depends on market dynamics to represent the market sentiments, and external drives for people to proceed with purchase.

$$\lambda_p = \begin{cases} \varepsilon_p, if\ \tau_p > \beta \\ 0, if\ \tau_p \leq \beta \end{cases} \quad (10)$$

In equation 10, model assures that only a fraction of intentions are converted to actual purchase. Once the purchase action is initiated, the η_p value for each agent is updated to appropriate subsection based on the result from equation 10.

2.3.7 Simulation

Simulation starts with a neutral opinion towards all the products and pre-set demographic distribution for all the agents. Every event that passes simulation timeline alters opinions based on the values defined by equations 5 and 6. Events in the simulation may also alter demographic distribution based on the type of event.

All events are constructed from the time the simulation is considered started with the duration of event. Every event is expected to have some preamble, peak and trail effect. These properties for each event are defined to be based on the event type to represent the event as close to the real life as possible. An event with possible market hype is expected to start with long preamble with slow rate of rise. An event with long lasting effect such as fuel price increase may have long trail time.

In order to quantify the event's effect before and after its occurrence, a parameter called event coefficient ρ^E is defined. The parameter ρ^E is always normalised to 1, and it assumes a fractional value between 0 and 1. The value 0 indicates no effect of the event, and value 1 indicates maximum effect of the event at the peak time.

Event timeline is constructed from the values of ρ^E for all the events. The timeline assumes 0 whenever event has no effect.

A fractional value for the event coefficient means that full effect of the event is not felt by the market. This is represented in the simulation as in equation 11 for the example case with event 1.

$${}_{\eta,\phi^{pr1}}^{E1}\Upsilon(new) = \rho^{E1} . {}_{\eta,\phi^{pr1}}^{E1}\Upsilon \quad (11)$$

It can be interpreted by equation 11 that only part of the event's effect is felt by the market in terms of alteration of agents' properties or opinions.

Simulation is run discretely at each time step which increments at minimum time quantum for the simulation. All the actions at the simulation time are updated and then results are calculated. Next time step of simulation takes values from event timeline and considers the parameter manipulation appropriately.

3 SIMULATION RESULTS

3.1 Plots

Plots usually represent the outcome of the simulation. In the present case, since market behaviour is of major concern, agents' opinions are plotted against each other for two products in the market.

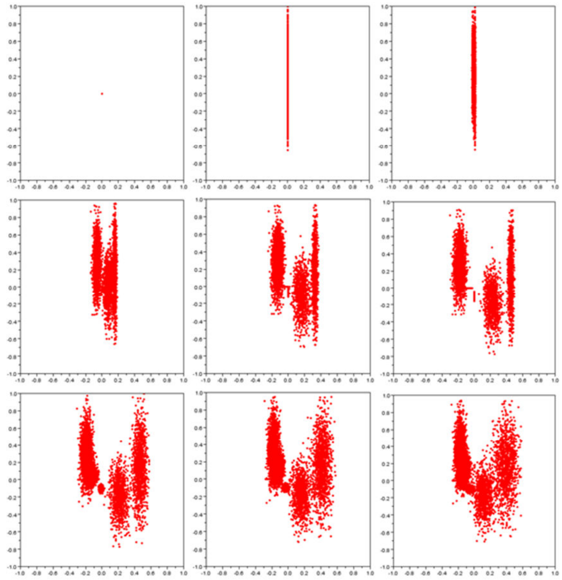


Figure 1 (a to i): Timeline view of opinions of agents for two products plotted against each other.

Figure 1 is a grid of 9 figures taken from different times of the simulation and presents the opinions of all agents for two products. Each dot in the plot window represents single agent. Figure 1 must be read horizontally for first row and same for second and third rows to trace with respect to timeline.

Each plot in figure 1 has four quadrants, while first quadrant represents the agents who prefer both the products, second quadrant represents the agents who prefer only product plotted on x-axis, and so on.

We can notice clusters of opinions formed with progress in simulation and concentration or dilution of those clusters. This actually represents the opinions formed about the customer segments and alteration of their opinions. The plots also represent the behaviour of prospective customers towards product preference. These results give insight into which market actions lead to the observed opinion drift.

3.2 Results

Other results of interest from the simulation are the market preference and purchase action.

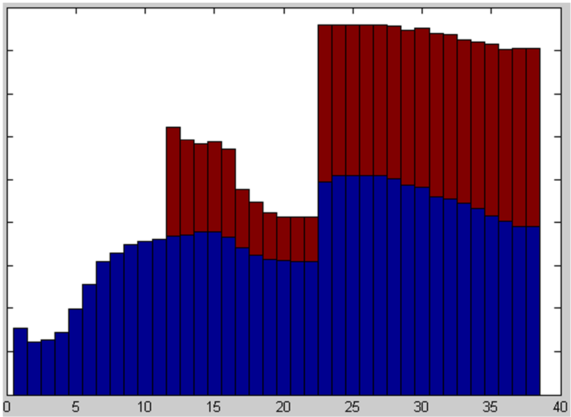


Figure 2: Timeline view of estimated sales in the market

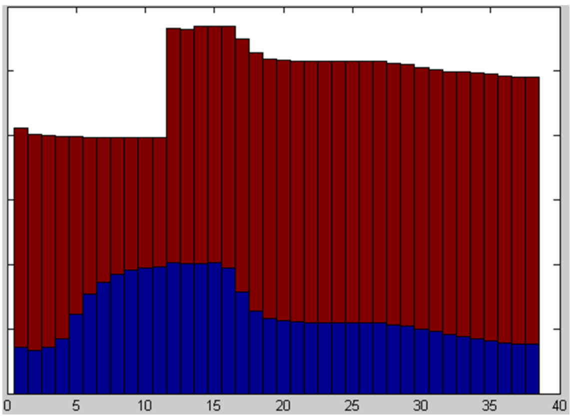


Figure 3: timeline view of product preference in the market

Figure 2 represents estimated sales variation with time and figure 3 represents preference between products 1 and 2 across all agents. As discussed earlier, product preference directly does not convert to sales. Intent to purchase and also market drive affect the final sales figures.

4 TUNING THE MODEL

Model presented in this paper shall work for all the theoretical purposes, while this model needs to be tuned to exactly reflect the real world. Tuning empathises that the effect of past events is appropriately considered.

Tuning process involves collecting data regarding past events, identifying the effected population, and identifying the final effect on sales. Once each event's effect is isolated and quantified, the same values are fed to the model to simulate and result values are checked against actual sales. User needs to continue tuning the model parameters till reasonable matching is achieved.

5 CONCLUSIONS

Our work on modelling market behaviour has been presented in this paper. Further, tuning based on market events and market results needs to be done to make the model ready for use. Once tuned, this model can be used to simulate for all the business plans. This model can be extended to many other avenues where consumers with diverse background are involved.

REFERENCES

C M Macal, M J North, 2010. 'Tutorial on agent-based modelling and simulation', *Journal of Simulation*, no. 4, pp. 151-162.

Michael J North, et al., 2010, 'multi scale agent based consumer market modelling', *Complexity*, vol. 15, issue 5, pp. 37-47

Paul Twomey, Richard Cadman, 2002, 'Agent based modelling of customer behaviour in the telecoms and media markets', info, vol. 4, iss. 1, pp. 56-63

Census 2011, Government of India

APPENDIX

η_p	Need group identification for agent p
α_p	Affordability group identification for agent p
τ_p	Intent to purchase for agent p
ϕ_p^{pr1}	Opinion for agent p towards product 1
${}_{\eta,\phi^{pr2}}^{E1}\Upsilon$	An event with identification number E1, whose effect is dependent on sub groups of need, and effects opinion of product 2
$\xi_{p,pr1}^{E1}$	The variation of opinion of agent p due to event E1
${}_{pr1,\eta^s}^{E1}\mu$	Mean of change of opinion for product 1 due to event E1, for the s^th^ sub group of need.
${}_{pr1,\eta^s}^{E1}\sigma$	Standard change of variation of opinion for product 1 due to event E1, for the s^th^ sub group of need.
ε_p	Identification of preferred product for the agent p
λ_p	Sales action of agent p
β	Purchase threshold for the market model
ρ^E	Event coefficient for the event E

SCILAB CODE

```
//Can be considered as Version 1.05//this file uses following functions // raghu_assignINDrule         v1.02// raghu_assignMIXrule1deep      v1.00//
raghu_hist   v1.09// raghu_SELECTIVEfeeling     v1.05// raghu_feeling              v1.01// raghu_hist_redraw           v2.01
funcprot(0);clear all;
function result=raghu_assignINDrule(inputcol, rulevalue, rulepercentage) //consider as version 1.02
   rulepercentage1=rulepercentage;
   for i=2:length(rulepercentage)
      rulepercentage1(i)=rulepercentage(i)+rulepercentage1(i-1);
   end //rulepercentage1 has accumulated values.for calculation purposes
   ravi=[(1:length(inputcol))',inputcol]; //ravi has index and random value
   [raja,komal]=gsort(ravi(:,2),'g','d'); // raja has sorted elements, komal has index order wrt original array
   raja=ravi(komal,:);// raja has sorted form of raja, but indices are jumbeed to represent order in original raja
   rulepercentage1=rulepercentage1.*(length(inputcol)/rulepercentage1($)); //count generated for each percentage
   rulepercentage1($)=length(inputcol); // added to avoid fractions omit one
   lastindex=1;
   for i=1:length(rulepercentage1)
      presentindex=floor(rulepercentage1(i));       raja((lastindex:presentindex),2)= rulevalue(i);        lastindex=presentindex+1;
   end
   [ravi,komal]=gsort(raja(:,1),'g','i');    ravi=raja(komal,:);    result=ravi(:,2);
endfunction
function result=raghu_assignMIXrule1deep(inputcol, rulevalue, rulepercentage, previousrulevalue, previousrulecol) //consider as version 1.00
   [m,p]=size(rulepercentage);//dependent percentage values
   [p,q]=size(previousrulevalue); //independent previous rule
   if m==q then//valid cross rule
      ravi=[(1:length(inputcol))',previousrulecol,inputcol]; //ravi has index, independent column, and dependent column
      for i=1:length(previousrulevalue) //for al the independent rule values,
         b=ravi(find(ravi(:,2)==previousrulevalue(i)),:); //b has all rows  that match with independent rule in independent rule column
         b(:,3)=raghu_assignINDrule(b(:,3),rulevalue,rulepercentage(i,:));//b's dependent rule column is assigned rule based on the value as mentioned
in the percentage row
         ravi(b(:,1),3)=b(:,3);//b is again inserted in to original ravi matrix in its own position.
      end
      result=ravi(:,3);
   else       //not a valid cross rule
      result=inputcol;       // not processed
   end
endfunction
function result=raghu_feeling(inputcol, v1, v2, type1)    //can be considered as version 1.01
   val1=v1;    mean1=v1;    val2=v2;    var1=v2;    AssignUnif=101;    AssignNorm=102;    ShiftUnif=201;    ShiftNorm=202;
   if(type1<200) then// assignment
      if (type1 == AssignUnif)  then
            result=grand(length(inputcol),1,'unf',val1,val2);
      elseif  type1 == AssignNorm    then
            result=grand(length(inputcol),1,'nor',mean1,var1);
      end
   elseif (type1<300) then//shift
      if type1 == ShiftUnif
            result=inputcol+grand(length(inputcol),1,'unf',val1,val2);
      elseif  type1 == ShiftNorm
            result=inputcol+grand(length(inputcol),1,'nor',mean1,var1);
      end
   else
      result=inputcol.*0;
   end
endfunction
function result=raghu_SELECTIVEfeeling(inputcol, controlcol, controlval, controlmean, controlvar, type1, xmin, xmax, EventTimeLine) // version 1.05
   if(EventTimeLine>0)
      if(EventTimeLine<1)
         controlmean=controlmean.*EventTimeLine;           controlvar=controlvar.*EventTimeLine;
      end
      if (unique(controlcol)==unique(controlval)')  //if all values from controlling column are covered and matching,
         a=length(controlval);           b=length(inputcol); //number of control vars //number of elements in coltrol col
         ram=[(1:b)' inputcol controlcol]; rahim=[];//ram: index, inputcol, control col
         for i=1:a//for all elements in control vals,
            hari=find(ram(:,3)==controlval(i)); ravan=ram(hari,:); //hari has all element indexes with control var 1,ravan has all rows with control var i
            if(controlvar(i)==0)
```

```
            else
              ravan(:,2)=raghu_feeling(ravan(:,2),controlmean(i),controlvar(i),type1); // ravan now has assigned feelings
            end
            rahim=[rahim;ravan];//rahim has the matrix building up.
          end
          [gopi,harsha]=gsort(rahim(:,1),'g','i'); // based on oroginal indices
          result=rahim(harsha,2);          result(find(result<xmin))=xmin;          result(find(result>xmax))=xmax;
      else
          result=inputcol;
      end
   else
      result = inputcol;
   end
endfunction
tic//clc
population=1e4;HistSlots=100;xmax=1;xmin=-xmax;HistCol=2; HistBorder=5; //Blue //RED
AssignUnif=101;AssignNorm=102;ShiftUnif=201;ShiftNorm=202;dT=0.1;PlotmodeYlim=1;PlotmodeYauto=0;ScalemodePer=1;
ScalemodeLin=0;Plotmode=PlotmodeYauto;AgingFactor=1-0.005;PurchaseThreshold=.7;i=getdate();
filename1 = 'RAGHU_BLR_'+string(i(1))+'_'+string(i(2))+'_'+string(i(6))+'_'+string(i(7))+'-'+string(i(8))+'-'+string(i(9))+'.gif';
demog.age=      grand(population,1,'nor',0,0.1);demog.gender=  grand(population,1,'nor',0,0.1);demog.afford=   grand(population,1,'nor',0,0.1);
demog.need=     grand(population,1,'nor',0,0.1);demog.intent=  grand(population,1,'nor',0,0);demog.action=   grand(population,1,'nor',0,0.1);
opinion.prod1= grand(population,1,'nor',0,0);opinion.prod2=   grand(population,1,'nor',0,0);opinion.prod3= grand(population,1,'nor',0,0);
opinion.pref=   grand(population,1,'nor',0,0);
age.rule=[61,62,63,64,65,66];age.Roffset=60;age.populate=[26.7,9.9,11.6,19.6,19.2,13.0];// age 1)<15 2)15-20  3)20-25  4)25-35  5)35-50  6)>50
gender.rule=[21,22];gender.Roffset=20;gender.age.populate=[51.1,48.9;51.7,48.3;53.2,46.8;52.6,47.4;53.8,46.2;52.4,47.6];// gender: 1)male 2) female
afford.rule=[51,52,53,54,55];afford.Roffset=50;afford.age.populate=[100,0,0,0,0;66.3,33.1,0.6,0,0;46.7,42.6,2.5,8.2,0;0,44.3,43.7,12.0,0;45.1,43.2,3.6,0.
9,7.2; 60.2,26.6,8.5,0.9,3.8];// 1)cant 2)with comprromise 3)with ease 4)as an option //5)wont consider
need.rule=[41,42,43,44];need.Roffset=40;need.age.populate=[100,0,0,0;45,30,20,5;20,30,30,20;5,30,30,35;30,20,30,20;20,55,20,5];// groups 1)no
need 2)optional 3)required 4)compulsory
for i=1:length(need.rule)
   for j=1:length(afford.rule)
      intent.grid(i,j)=(need.rule(i)-need.Roffset-1)*(afford.rule(j)-afford.Roffset-1);
   end
end
intent.grid=intent.grid/(mean(need.rule-need.Roffset-1)*mean(afford.rule-afford.Roffset-1));
demog.age=      raghu_assignINDrule(demog.age,age.rule,age.populate);
demog.gender=   raghu_assignMIXrule1deep(demog.gender,gender.rule,gender.age.populate,age.rule,demog.age);
demog.afford=   raghu_assignMIXrule1deep(demog.afford,afford.rule,afford.age.populate,age.rule,demog.age);
demog.need=     raghu_assignMIXrule1deep(demog.need,need.rule,need.age.populate,age.rule,demog.age);
intent.hist.xmin=0;intent.hist.xmax=max(max(intent.grid));
intent.hist.slots=13;pref.hist.xmin=0;pref.hist.xmax=3;pref.hist.slots=4;
action.hist.xmin=1;action.hist.xmax=3;action.hist.slots=3;
opinion.prod1=opinion.prod1.*0;opinion.prod2=opinion.prod2.*0;opinion.prod3=opinion.prod3.*0;
//events to be defined, and their durations, and their effects
events.type=[ 1, 2,3,4,5 ];events.starttick=[1,12,28,23,23]; //Nth day of occurance
events.peaktick=[5, 5,6,1,1];events.effectiveticks=[ 15,12,10,1,1]; //kth tick after start which is the peak.//effect duration of each event
events.decaycoeff=[5,10,3,1,1]; // decay coeff for esponential decay
NofEtype=length(unique(events.type));EventScale=zeros(NofEtype,(max(events.starttick+events.effectiveticks)));
//exponential decay
for i=1:length(events.type)
   EventScale(events.type(i),(events.starttick(i)+events.peaktick(i)):(events.starttick(i)-1+events.effectiveticks(i)))=
EventScale(events.type(i),(events.starttick(i)+events.peaktick(i)):(events.starttick(i)-1+events.effectiveticks(i))) + exp(-
events.decaycoeff(i).*(0:events.effectiveticks(i)-events.peaktick(i)-1)/(events.effectiveticks(i)));
   EventScale(events.type(i),(events.starttick(i)):(events.starttick(i)+events.peaktick(i)-1))=
EventScale(events.type(i),(events.starttick(i)):(events.starttick(i)+events.peaktick(i)-1)) + exp(-events.decaycoeff(i).*((events.peaktick(i)-1) :-1
:0)/(events.peaktick(i)));
end// figure(998); stairs(0.5:1:max(events.starttick+events.effectiveticks)-0.5,EventScale'); xlim([1,max(events.starttick+events.effectiveticks)])
prod1.Arule.age.mean=[0.200,0.100,0.050,0.000,0.100 0.300];prod1.Arule.age.SD=[ 0.000,0.000,0.000,0.000,0.000,0.000];prod1.ID=1;
prod2.Arule.age.mean=[0.100,0.100,0.200,0.300,0.700,0.000];prod2.Arule.age.SD=[0.000,0.200,0.300,0.200,0.000,0.000];prod2.ID=2;
prod3.Arule.age.mean=[0.100,0.200,0.700,0.300,0.200,0.300];prod3.Arule.age.SD= [0.000,0.000,0.000,0.000,0.000,0.000];prod3.ID=3;
// before 2000.. prod2 range has most popularity.
prod1.age1.mean=[0.000,0.050,0.100,-.040,-.030,-.050];prod1.age1.SD=[0.000,0.030,0.010,0.020,0.020,0.010];
prod2.age1.mean=[-.022,-.061,-.011,0.001,0.002,0.002];prod2.age1.SD=[0.010,0.020,0.002,0.000,0.000,0.000];
prod3.age1.mean=[0.020,0.080,0.050,-.010,-.020,-.040];prod3.age1.SD=[0.000,0.000,0.000,0.000,0.000,0.000];
//2000- launch of prod1. first 150 cc bike,
```

```
prod1.age2.mean=[-.010,-.020,-.010,0.000,0.000,0.000];prod1.age2.SD=[0.010,0.020,0.040,0.000,0.000,0.000];
prod2.age2.mean=[0.000,0.000,0.020,0.040,0.050,0.000];prod2.age2.SD=[0.000,0.000,0.100,0.050,0.060,0.000];
prod3.age2.mean=[0.020,0.080,0.050,-.010,-.020,-.040];prod3.age2.SD=[0.010,0.010,0.020,0.040,0.030,0.010];
//2001 - launch of prod2. youngsters are unchanged. elders are happy about//the mileage, loooks, etc. //2001-launch of prod3 too. sports bike.
prod1.age3.mean=[-.010,-.020,-.010,0.000,0.000,0.000];prod1.age3.SD=[0.010,0.020,0.040,0.000,0.000,0.000];
prod2.age3.mean=[0.000,0.000,0.020,0.040,0.050,0.000];prod2.age3.SD=[0.000,0.000,0.000,0.000,0.000,0.000];
prod3.age3.mean=[0.020,0.040,0.020,-.010,-.020,-.040];prod3.age3.SD=[0.010,0.010,0.020,0.040,0.030,0.010];
//2004 launch of prod2
need.shift1.mean=[0.000,0.000,1.000,0.000,0.000,0.000];need.shift1.SD=[1.000,2.000,3.000,1.000,1.000,1.000];
afford.shift1.mean=[0.000,0.000,1.000,0.000,0.000,0.000];afford.shift1.SD=[1.000,2.000,3.000,1.000,1.000,1.000];  // alteration of need
  drawlater();//// Plotting
scf(999);clf;
subplot(2,6,1);   bins=[];   x123=linspace(xmin,xmax,HistSlots);   data=opinion.prod1;
  for i=1:length(x123)-1
    bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
  end
  bins(HistSlots)=length(find(data>=x123(HistSlots)));   bins=bins.*100/population;   bar(x123,bins,1);   h11=gca();
  qwe=h11.x_ticks;qwe(2)=[-1; 0; 1];qwe(3)=['NO' ;'neutral' ;'YES'];h11.x_ticks=qwe;
subplot(2,6,2);   bins=[];   x123=linspace(xmin,xmax,HistSlots);   data=opinion.prod2;
  for i=1:length(x123)-1
    bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
  end
  bins(HistSlots)=length(find(data>=x123(HistSlots)));   bins=bins.*100/population;   bar(x123,bins,1);   h12=gca();
  qwe=h12.x_ticks;qwe(2)=[-1; 0; 1];qwe(3)=['NO' ;'neutral' ;'YES'];h12.x_ticks=qwe;
subplot(2,6,3);   bins=[];   x123=linspace(xmin,xmax,HistSlots);   data=opinion.prod3;
  for i=1:length(x123)-1
    bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
  end
  bins(HistSlots)=length(find(data>=x123(HistSlots)));   bins=bins.*100/population;   bar(x123,bins,1);   h13=gca();
  qwe=h13.x_ticks;qwe(2)=[-1; 0; 1];qwe(3)=['NO' ;'neutral' ;'YES'];h13.x_ticks=qwe;
subplot(2,6,4);   bins=[];   x123=linspace(intent.hist.xmin,intent.hist.xmax,intent.hist.slots);   data=demog.intent;
  for i=1:length(x123)-1
    bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
  end
  bins(intent.hist.slots)=length(find(data>=x123(intent.hist.slots)));   bins=bins.*100/population;   bar(x123,bins,1);   h14=gca();
  qwe=h14.x_ticks;qwe(2)=[0; 2; 4];qwe(3)=['NO' ;'medium' ;'high'];h14.x_ticks=qwe;
subplot(2,6,5);   bins=[];   x123=linspace(pref.hist.xmin,pref.hist.xmax,pref.hist.slots);   data=opinion.pref;
  for i=1:length(x123)-1
    bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
  end
  bins(pref.hist.slots)=length(find(data>=x123(pref.hist.slots)));   bins=bins.*100/population;   bar(x123,bins,1);   h15=gca();
  qwe=h15.x_ticks;qwe(2)=[0; 1; 2;3];qwe(3)=['none' ;'prod1' ;'prod2' ;'prod3'];h15.x_ticks=qwe;
subplot(2,6,6);   bins=[];   x123=linspace(action.hist.xmin,action.hist.xmax,action.hist.slots);   data=demog.action;
  for i=1:length(x123)-1
    bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
  end
  bins(action.hist.slots)=length(find(data>=x123(action.hist.slots)));   bar(x123,bins,1);   h16=gca();
  qwe=h16.x_ticks;qwe(2)=[1; 2;3];qwe(3)=['prod1' ;'prod2' ;'prod3'];h16.x_ticks=qwe;
  subplot(2,3,4);   plot(opinion.prod1,opinion.prod2,'r.');
  subplot(2,3,5);   plot(opinion.prod2,opinion.prod3,'r.');
  subplot(2,3,6);   plot(opinion.prod3,opinion.prod1,'r.');
  myF=gcf();
  myF.children(9).children.children.foreground=2;   myF.children(8).children.children.foreground=2;
  myF.children(7).children.children.foreground=2;   myF.children(6).children.children.foreground=5;
  myF.children(5).children.children.foreground=5;   myF.children(4).children.children.foreground=5;
  myF.children(4).data_bounds(:,2)=[0;population];   myF.children(3).children.children.mark_size=2;
  myF.children(2).children.children.mark_size=2;   myF.children(1).children.children.mark_size=2;
 drawnow();//// TimeLine//realtimeinit(0.2);//realtime(0);
opinion.prod2=raghu_SELECTIVEfeeling(opinion.prod2,demog.age,age.rule,prod2.Arule.age.mean,prod2.Arule.age.SD,ShiftNorm,xmin,xmax,1);
for z=1:max((events.starttick+events.effectiveticks))
   //realtime(i);   // keep backup tmp vars to process fro incremental changes
  demog.Ntmp=demog.need;    demog.Atmp=demog.afford;
  //time events
  opinion.prod1= raghu_SELECTIVEfeeling(opinion.prod1,  demog.age,  age.rule,   prod1.age1.mean,   prod1.age1.SD,   ShiftNorm,xmin,  xmax,
EventScale(1,z));
```

```
   opinion.prod2=  raghu_SELECTIVEfeeling(opinion.prod2,  demog.age, age.rule,  prod2.age1.mean,   prod2.age1.SD,   ShiftNorm,xmin, xmax,
EventScale(1,z));
   opinion.prod1= raghu_SELECTIVEfeeling(opinion.prod1, demog.age, age.rule,  prod1.age2.mean,  prod1.age2.SD,  ShiftNorm,xmin, xmax,
EventScale(2,z));
   opinion.prod2=  raghu_SELECTIVEfeeling(opinion.prod2,  demog.age, age.rule,  prod2.age2.mean,   prod2.age2.SD,   ShiftNorm,xmin, xmax,
EventScale(2,z));
   opinion.prod3= raghu_SELECTIVEfeeling(opinion.prod3, demog.age, age.rule,  prod3.age2.mean,  prod3.age2.SD,  ShiftNorm,xmin, xmax,
EventScale(2,z));
   opinion.prod1= raghu_SELECTIVEfeeling(opinion.prod1, demog.age, age.rule,  prod1.age3.mean,  prod1.age3.SD,  ShiftNorm,xmin, xmax,
EventScale(3,z));
   opinion.prod2=  raghu_SELECTIVEfeeling(opinion.prod2,  demog.age, age.rule,  prod2.age3.mean,   prod2.age3.SD,   ShiftNorm,xmin, xmax,
EventScale(3,z));
   opinion.prod3= raghu_SELECTIVEfeeling(opinion.prod3, demog.age, age.rule,  prod3.age3.mean,  prod3.age3.SD,  ShiftNorm,xmin, xmax,
EventScale(3,z));
   demog.Ntmp=   raghu_SELECTIVEfeeling(demog.need,   demog.need, need.rule, need.shift1.mean,  need.shift1.SD,  ShiftNorm,min(need.rule),
max(need.rule),  EventScale(4,z));
   demog.Atmp=   raghu_SELECTIVEfeeling(demog.afford,   demog.afford, afford.rule, afford.shift1.mean,  afford.shift1.SD,
ShiftNorm,min(afford.rule), max(afford.rule),  EventScale(5,z));
  demog.intent=((demog.Ntmp-need.Roffset-1).*(demog.Atmp-afford.Roffset-1))/(mean(need.rule-need.Roffset-1)*mean(afford.rule-afford.Roffset-1));
// change of intention to buy for population
   a=max(opinion.prod1,max(opinion.prod2,opinion.prod3)); // update preference for population
   opinion.pref(find(a==opinion.prod1))=prod1.ID;   opinion.pref(find(a==opinion.prod2))=prod2.ID;   opinion.pref(find(a==opinion.prod3))=prod3.ID;
   opinion.pref(find(a==0))=0;
   demog.action(:,1)=0;   demog.action(find(demog.intent>PurchaseThreshold))=opinion.pref(find(demog.intent>PurchaseThreshold)); //update action
   demog.need=   round(demog.Ntmp);   demog.afford=  round(demog.Atmp); // update the new values
   opinion.prod1= AgingFactor*opinion.prod1; opinion.prod2= AgingFactor*opinion.prod2; opinion.prod3= AgingFactor*opinion.prod3; //memory fades
   drawlater();   //plotting
   bins=[];   x123=linspace(xmin,xmax,HistSlots);   data=opinion.prod1;
   for i=1:length(x123)-1
     bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
   end
   bins(HistSlots)=length(find(data>=x123(HistSlots)));   bins=bins.*100/population;myF.children(9).children.children.data(:,2)=bins;
   bins=[];   x123=linspace(xmin,xmax,HistSlots);   data=opinion.prod2;
   for i=1:length(x123)-1
     bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
   end
   bins(HistSlots)=length(find(data>=x123(HistSlots)));   bins=bins.*100/population;myF.children(8).children.children.data(:,2)=bins;
   bins=[];   x123=linspace(xmin,xmax,HistSlots);   data=opinion.prod3;
   for i=1:length(x123)-1
     bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
   end
   bins(HistSlots)=length(find(data>=x123(HistSlots)));   bins=bins.*100/population;myF.children(7).children.children.data(:,2)=bins;
   bins=[];   x123=linspace(intent.hist.xmin,intent.hist.xmax,intent.hist.slots);   data=demog.intent;
   for i=1:length(x123)-1
     bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
   end
   bins(intent.hist.slots)=length(find(data>=x123(intent.hist.slots)));   bins=bins.*100/population;myF.children(6).children.children.data(:,2)=bins;
   bins=[];   x123=linspace(pref.hist.xmin,pref.hist.xmax,pref.hist.slots);   data=opinion.pref;
   for i=1:length(x123)-1
     bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
   end
   bins(pref.hist.slots)=length(find(data>=x123(pref.hist.slots)));   bins=bins.*100/population;myF.children(5).children.children.data(:,2)=bins;
   bins=[];   x123=linspace(action.hist.xmin,action.hist.xmax,action.hist.slots);   data=demog.action;
   for i=1:length(x123)-1
     bins(i)=length(find((data>=x123(i))&(data<x123(i+1))));
   end
   bins(action.hist.slots)=length(find(data>=x123(action.hist.slots)));myF.children(4).children.children.data(:,2)=bins;
myF.children(3).children.children.data(:,1)=opinion.prod1;myF.children(3).children.children.data(:,2)=opinion.prod2;
myF.children(2).children.children.data(:,1)=opinion.prod2;myF.children(2).children.children.data(:,2)=opinion.prod3;
myF.children(1).children.children.data(:,1)=opinion.prod3;myF.children(1).children.children.data(:,2)=opinion.prod1;
   drawnow();  myF.children(2).title.text=string(z);
end
disp(toc());disp('done');
```

END SCILAB CODE

MATLAB CODE

```
%% header%Can be considered as Version 1.05%this file uses following functions% raghu_assignINDrule          v1.02% raghu_assignMIXrule1deep
v1.00% raghu_hist                    v1.09% raghu_SELECTIVEfeeling          v1.05% raghu_feeling                v1.01% raghu_hist_redraw             v2.01
clear all;tic;cd 'E:\BPLG\MAT';clc;population=1e4;HistSlots=100;xmax=1;xmin=-xmax;HistCol='BLUE';HistBorder='RED';
AssignUnif=101;AssignNorm=102;ShiftUnif=201;ShiftNorm=202;dT=0.1;PlotmodeYlim=1;PlotmodeYauto=0;ScalemodePer=1;ScalemodeLin=0;
Plotmode=PlotmodeYauto;AgingFactor=1-0.005;PurchaseThreshold=.7;i=clock; GenerateGIF=1;
filename1 = strcat('RAGHU_BLR_',num2str(i(1)),'_',num2str(i(2)),'_',num2str(i(3)),'_',num2str(i(4)),'-',num2str(i(5)),'-',num2str(i(6)),'.gif');
%%mcc -m BLR* to generate .exe and dependencies. %pcode *.m to generate pcode files.
demog.age=      random('normal',    0,  0.1,    population,1);demog.gender=   random('normal',    0,  0.1,    population,1);
demog.afford=   random('normal',    0,  0.1,    population,1);demog.need=     random('normal',    0,  0.1,    population,1);
demog.intent=   random('normal',    0,  0,      population,1);demog.action=   random('normal',    0,  0,      population,1);
opinion.prod1= random('uniform',   0,  0,      population,1);opinion.prod2=   random('uniform',   0,  0,      population,1);
opinion.prod3= random('uniform',   0,  0,      population,1);opinion.pref=    random('uniform',   0,  0,      population,1);
%% demographics%rules to distribute demographics% XXX.rule=[A B C D] % XXX.populate=[p q r s]% A,B,C,D are properties
% p+q+r+s = 100, and each is percentage of people distribution
age.rule= [61, 62, 63, 64, 65, 66]; age.Roffset=60; age.populate= [26.7, 9.9, 11.6, 19.6, 19.2, 13.0];% 1)<15 2)15-20  3)20-25  4)25-35  5)35-50  6)>50
gender.rule= [21, 22]; gender.Roffset=20; gender.age.populate= [51.1, 48.9; 51.7, 48.3; 53.2, 46.8; 52.6, 47.4; 53.8, 46.2; 52.4, 47.6];%: 1)male 2) female
afford.rule= [51, 52, 53, 54, 55];afford.Roffset=50; % groupts 1)cant 2)with comprromise 3)with ease 4)as an option %5)wont consider
afford.age.populate=[100, 0, 0 , 0, 0;66.3, 33.1, 0.6 , 0 , 0;46.7,42.6,2.5, 8.2,  0; 0,   44.3,43.7,12.0, 0; 45.1,43.2,3.6, 0.9, 7.2;60.2,26.6,8.5,0.9,3.8];
need.rule=[ 41 ,42,43,44];need.Roffset=40; % groups 1)no need 2)optional 3)required 4)compulsory
need.age.populate=[100 ,0,0,0; 45,30,20,5; 20,30,30,20; 5, 30,30,35; 30, 20,30,20; 20,55,20,5];
for i=1:length(need.rule)
   for j=1:length(afford.rule)
      intent.grid(i,j)=(need.rule(i)-need.Roffset-1)*(afford.rule(j)-afford.Roffset-1);
   end
end
intent.grid=intent.grid/(mean(need.rule-need.Roffset-1)*mean(afford.rule-afford.Roffset-1));
demog.age=      raghu_assignINDrule(demog.age,age.rule,age.populate);
demog.gender=   raghu_assignMIXrule1deep(demog.gender,gender.rule,gender.age.populate,age.rule,demog.age);
demog.afford=   raghu_assignMIXrule1deep(demog.afford,afford.rule,afford.age.populate,age.rule,demog.age);
demog.need=     raghu_assignMIXrule1deep(demog.need,need.rule,need.age.populate,age.rule,demog.age);
intent.hist.xmin=0;intent.hist.xmax=max(max(intent.grid));intent.hist.slots=13;pref.hist.xmin=0;pref.hist.xmax=3;pref.hist.slots=4;
action.hist.xmin=1;action.hist.xmax=3;action.hist.slots=3;
opinion.prod1=opinion.prod1.*0;opinion.prod2=opinion.prod2.*0;opinion.prod3=opinion.prod3.*0; %% init distribute
%% events%events to be defined, and their durations, and their effects
events.type=[ 1, 2,3, 4,5 ];events.starttick=[  1, 12,28,23,23]; %Nth day of occurance
events.peaktick=[ 5 ,5,6,1 , 1]; events.effectiveticks=[ 15, 12 ,10,1 , 1];%effect duration of each event %kth tick after start which is the peak.
events.decaycoeff=[5, 10 ,3, 1 , 1]; % decay coeff for esponential decay
NofEtype=length(unique(events.type));EventScale=zeros(NofEtype,(max(events.starttick+events.effectiveticks)));
%exponential decay
for i=1:length(events.type)
   EventScale(events.type(i),(events.starttick(i)+events.peaktick(i)):(events.starttick(i)-1+events.effectiveticks(i)))=
EventScale(events.type(i),(events.starttick(i)+events.peaktick(i)):(events.starttick(i)-1+events.effectiveticks(i))) + exp(-
events.decaycoeff(i).*(0:events.effectiveticks(i)-events.peaktick(i)-1)/(events.effectiveticks(i)));
   EventScale(events.type(i),(events.starttick(i)):(events.starttick(i)+events.peaktick(i)-1))=
EventScale(events.type(i),(events.starttick(i)):(events.starttick(i)+events.peaktick(i)-1)) + exp(-events.decaycoeff(i).*((events.peaktick(i)-1) :-1
:0)/(events.peaktick(i)));
end
% figure(998); stairs(0.5:1:max(events.starttick+events.effectiveticks)-0.5,EventScale'); xlim([1,max(events.starttick+events.effectiveticks)])
%% Shift-Assign Rules%rules to define opinion shift patterns and directions with each event
Prod1.Arule.age.mean= [0.200, 0.100, 0.050, 0.000, 0.100, 0.300]; prod1.Arule.age.SD= [0.000, 0.000, 0.000, 0.000, 0.000, 0.000]; prod1.ID=1;
prod2.Arule.age.mean= [0.100, 0.100, 0.200, 0.300, 0.700, 0.000]; prod2.Arule.age.SD= [0.000, 0.200, 0.300, 0.200, 0.000, 0.000]; prod2.ID=2;
prod3.Arule.age.mean= [0.100, 0.200, 0.700, 0.300, 0.200, 0.300]; prod3.Arule.age.SD= [0.000, 0.000, 0.000, 0.000, 0.000, 0.000]; prod3.ID=3;
% before 2000. this is the time of mileage bikes. prod2range has most% popularity.
Prod1.age1.mean= [0.000, 0.050, 0.100,-.040,-.030,-.050   ]; prod1.age1.SD= [0.000, 0.030, 0.010, 0.020, 0.020, 0.010   ];
prod2.age1.mean=  [-.022 ,-.061,-.011,0.001,0.002,0.002   ]; prod2.age1.SD= [0.010,0.020,0.002,0.000,0.000,0.000   ];
prod3.age1.mean= [0.020, 0.080, 0.050,-.010,-.020,-.040   ]; prod3.age1.SD= [0.000, 0.000, 0.000, 0.000, 0.000, 0.000   ];
%2000- launch of prod1. first 150 cc bike, youth attraction. %only youth liked it for the adriniline feel. elders disliked it for the%rude ride. kids dmired
it. infants have neutral opinion. %popularity for prod2remained same in adults, youngsters started to dislike%it for the looks and lack of power.
Prod1.age2.mean= [-.010,-.020,-.010, 0.000, 0.000, 0.000   ];Prod1.age2.SD= [0.010, 0.020, 0.040, 0.000, 0.000, 0.000   ];
prod2.age2.mean= [0.000, 0.000, 0.020, 0.040, 0.050, 0.000   ];prod2.age2.SD= [0.000, 0.000, 0.100, 0.050, 0.060, 0.000   ];
prod3.age2.mean= [0.020, 0.080, 0.050,-.010,-.020,-.040   ];prod3.age2.SD= [0.010, 0.010, 0.020, 0.040, 0.030, 0.010   ];
%2001 - launch of prod2. youngsters are unchanged. elders are happy about%the mileage, loooks, etc. %2001-launch of prod3 too. sports bike. Prod1
opinions split because of%options available.
Prod1.age3.mean= [-.010,-.020,-.010,0.000,0.000,0.000   ];Prod1.age3.SD= [0.010,0.020,0.040,0.000,0.000, 0.000   ];
```

```
prod2.age3.mean= [0.000,0.000,0.020,0.040,0.050,0.000   ]; prod2.age3.SD= [0.000,0.000,0.000,0.000,0.000,0.000   ];
prod3.age3.mean= [0.020,0.040,0.020,-.010,-.020,-.040   ];prod3.age3.SD= [0.010,0.010,0.020,0.040,0.030,0.010   ];
%2004 launch of prod2
need.shift1.mean= [0.000,0.000,1.000,0.000,0.000,0.000   ];need.shift1.SD= [1.000,2.000,3.000,1.000, 1.000, 1.000   ];  % alteration of need
afford.shift1.mean= [0.000, 0.000, 1.000, 0.000, 0.000,0.000   ];afford.shift1.SD= [1.000,2.000, 3.000, 1.000,1.000 ,1.000   ];  % alteration of need
%% Plotting
figure(999);clf;subplot(2,6,1);h11=raghu_hist(opinion.prod1,  HistSlots, HistCol,HistCol, xmin, xmax, Plotmode,ScalemodePer,'opinion.prod1(%) ');
set(gca,'XTickLabel',{'NO' ;'neutral' ;'YES'},'XTick',[-1 0 1]);
subplot(2,6,2);h12=raghu_hist(opinion.prod2,    HistSlots, HistCol,HistCol, xmin, xmax, Plotmode,ScalemodePer,'opinion.prod2 (%) ');
set(gca,'XTickLabel',{'NO' ;'neutral' ;'YES'},'XTick',[-1 0 1]);
subplot(2,6,3);h13=raghu_hist(opinion.prod3,  HistSlots, HistCol,HistCol, xmin, xmax, Plotmode,ScalemodePer,'opinion.prod3(%) ');
set(gca,'XTickLabel',{'NO' ;'neutral' ;'YES'},'XTick',[-1 0 1]);subplot(2,6,4);
h14=raghu_hist(demog.intent, intent.hist.slots,  HistCol,HistBorder, intent.hist.xmin, intent.hist.xmax, Plotmode,ScalemodePer,'purchase intent(%)');
set(gca,'XTickLabel',{'NO' ;'medium' ;'high'},'XTick',[0 2 4]);
subplot(2,6,5);h15=raghu_hist(opinion.pref,pref.hist.slots,HistCol,HistBorder,pref.hist.xmin,pref.hist.xmax,Plotmode,ScalemodePer,'preference(%)');
set(gca,'XTickLabel',{'none' ;'prod1' ;'prod2' ;'prod3'},'XTick',[0 1 2 3]);
subplot(2,6,6);h16=raghu_hist(demog.action,action.hist.slots,HistCol,HistBorder,action.hist.xmin,action.hist.xmax,Plotmode,ScalemodeLin,'Sales(#)');
set(gca,'XTickLabel',{'none' ;'prod1' ;'prod2' ;'prod3'},'XTick',[0 1 2 3]);
subplot(2,6,7:8);h21=plot((opinion.prod1),(opinion.prod3),'r.');grid on;xlabel('prod1');xlim([-1 1]);ylim([-1 1]);text(-1.2,1.2,'prod3');
set(h21,'XDataSource','(opinion.prod1)','YDataSource','(opinion.prod3)'); %set(gca,'XTickLabel',{'NO' ;'neutral' ;'YES'},'XTick',[-1 0 1],'YTickLabel',{'NO'
;'neutral' ;'YES'},'YTick',[-1 0 1]);
subplot(2,6,9:10);h22=plot((opinion.prod3),(opinion.prod2),'r.');grid on;xlabel('prod3');xlim([-1 1]);ylim([-1 1]);text(-1.2,1.2,'prod2');
set(h22,'XDataSource','(opinion.prod3)','YDataSource','(opinion.prod2)');
subplot(2,6,11:12);h23=plot((opinion.prod2),(opinion.prod1),'r.');grid on;xlabel('prod2');xlim([-1 1]);ylim([-1 1]);text(-1.2,1.2,'prod1');
set(h23,'XDataSource','(opinion.prod2)','YDataSource','(opinion.prod1)');
subplot(2,6,9:10); % for title to change with every frame
if(GenerateGIF==1)
   im = frame2im(getframe(999));       [imind,cm] = rgb2ind(im,256);    imwrite(imind,cm,filename1,'gif', 'Loopcount',inf);
end
%% TimeLine%simulate for all ticks
opinion.prod2=raghu_SELECTIVEfeeling(opinion.prod2,demog.age,age.rule,prod2.Arule.age.mean,prod2.Arule.age.SD,ShiftNorm,xmin,xmax,1);
for i=1:max((events.starttick+events.effectiveticks))
   % keep backup tmp vars to process fro incremental changes
   demog.Ntmp=demog.need;      demog.Atmp=demog.afford;
   %time events
   opinion.prod1=raghu_SELECTIVEfeeling(opinion.prod1,demog.age,age.rule,prod1.age1.mean,   prod1.age1.SD,    ShiftNorm,xmin,  xmax,   EventScale(1,i));
   opinion.prod2=raghu_SELECTIVEfeeling(opinion.prod2,demog.age,  age.rule,prod2.age1.mean,  prod2.age1.SD,ShiftNorm,xmin,xmax,   EventScale(1,i));
   opinion. prod1=raghu_SELECTIVEfeeling(opinion.prod1,demog.age,age.rule,prod1.age2.mean,prod1.age2.SD, ShiftNorm,xmin,  xmax,   EventScale(2,i));
   opinion.prod2=raghu_SELECTIVEfeeling(opinion.prod2,demog.age,age.rule,prod2.age2.mean,prod2.age2.SD,ShiftNorm,xmin,  xmax,   EventScale(2,i));
   opinion.prod3=raghu_SELECTIVEfeeling(opinion.prod3,demog.age,age.rule,prod3.age2.mean,prod3.age2.SD,ShiftNorm,xmin,  xmax,   EventScale(2,i));
    opinion. prod1=raghu_SELECTIVEfeeling(opinion.prod1,demog.age,age.rule,prod1.age3.mean,prod1.age3.SD, ShiftNorm,xmin,  xmax,   EventScale(3,i));
   opinion.prod2=raghu_SELECTIVEfeeling(opinion.prod2,demog.age,age.rule,prod2.age3.mean,prod2.age3.SD, ShiftNorm,xmin, xmax,  EventScale(3,i));
   opinion.prod3= raghu_SELECTIVEfeeling(opinion.prod3,demog.age,age.rule,prod3.age3.mean,prod3.age3.SD,ShiftNorm,xmin,xmax,EventScale(3,i));
demog.Ntmp=     raghu_SELECTIVEfeeling(demog.need,     demog.need, need.rule,  need.shift1.mean,   need.shift1.SD,    ShiftNorm,min(need.rule),
max(need.rule),   EventScale(4,i));
   demog.Atmp=raghu_SELECTIVEfeeling(demog.afford,demog.afford,afford.rule,afford.shift1.mean,afford.shift1.SD,ShiftNorm,min(afford.rule),
max(afford.rule),EventScale(5,i));
   % change of intention to buy for population
 demog.intent = ((demog.Ntmp-need.Roffset-1).*(demog.Atmp-afford.Roffset-1))/(mean(need.rule-need.Roffset-1)*mean(afford.rule-afford.Roffset-1));
   % update preference for population
   a=max(opinion. prod1,max(opinion.prod2,opinion.prod3));    opinion.pref(find(a==opinion. prod1))= prod1.ID;
  opinion.pref(find(a==opinion.prod2))= prod2.ID;    opinion.pref(find(a==opinion.prod3))=prod3.ID;    opinion.pref(find(a==0))=0;
      %update action for population
   demog.action(:,1)=0;    demog.action(find(demog.intent>PurchaseThreshold))=opinion.pref(find(demog.intent>PurchaseThreshold));
      % update the new values
   demog.need=     round(demog.Ntmp);     demog.afford=   round(demog.Atmp);
      %memory fades with time
   opinion.prod1= AgingFactor*     opinion.prod1;    opinion.prod2=   AgingFactor*     opinion.prod2;    opinion.prod3= AgingFactor*     opinion.prod3;
   %plotting
  raghu_hist_redraw(opinion.prod1,HistSlots,xmin,xmax,h11,ScalemodePer);raghu_hist_redraw(opinion.prod2,HistSlots,xmin,xmax,h12,ScalemodePer);
   raghu_hist_redraw(opinion.prod3, HistSlots, xmin, xmax, h13, ScalemodePer);
  raghu_hist_redraw(demog.intent,intent.hist.slots,intent.hist.xmin,   intent.hist.xmax,h14,ScalemodePer);
  raghu_hist_redraw(opinion.pref,   pref.hist.slots,    pref.hist.xmin,     pref.hist.xmax,     h15,    ScalemodePer);
   raghu_hist_redraw(demog.action,   action.hist.slots,  action.hist.xmin,   action.hist.xmax,   h16,    ScalemodeLin);
   refreshdata(h21);    refreshdata(h22);    refreshdata(h23);    drawnow;    title(num2str(i));
```

```
   if(GenerateGIF==1)
      im = frame2im(getframe(999));          [imind,cm] = rgb2ind(im,256);       imwrite(imind,cm,filename1,'gif','WriteMode','append');
   end
   %pause(dT);
end%% end
disp(toc())disp('done')
```

```
function result=raghu_assignINDrule(inputcol,rulevalue,rulepercentage)
   % input col is the column with random values    %rulevalue is the value of rule. i.e. some number, which stands for assigned characteristic. this is simple array    %rule percentage is the percentage of total people in input column who have the characteristic. this is simple array    %values in rule percentage should add to 100.    %rulevalue and rule percentage should have same size.
   rulepercentage1=rulepercentage;
   for i=2:length(rulepercentage)
      rulepercentage1(i)=rulepercentage(i)+rulepercentage1(i-1);
   end %rulepercentage1 has accumulated values. just for calculation purposes
   ravi=[(1:length(inputcol))',inputcol]; % ravi has index and random value provided
   [raja,komal]=sort(ravi(:,2),'descend'); % raja has sorted elements, komal has index order wrt original array
   raja=ravi(komal,:);% raja has sorted form of raja, but indices are jumbeed to represent order in original raja
   rulepercentage1=rulepercentage1.*(length(inputcol)/rulepercentage1(end)); %count generated for each percentage
   rulepercentage1(end)=length(inputcol); %// added to avoid fractions omit one variable.
   lastindex=1;
   for i=1:length(rulepercentage1)
      presentindex=floor(rulepercentage1(i));
      raja((lastindex:presentindex),2)= rulevalue(i);  % values assigned
      lastindex=presentindex+1;
   end
   [ravi,komal]=sort(raja(:,1),'ascend');
   ravi=raja(komal,:);
   result=ravi(:,2);
```

```
function result = raghu_assignMIXrule1deep(inputcol,rulevalue,rulepercentage,previousrulevalue,previousrulecol)
%consider as version 1.00
%input column is the original column with random values. This property is%dependant on other property. (educational level) as in previousrulecol %rule value is a simple array, with values indicating a characteristic (1-none, 2-10th, 3-be,)    %rule percentage is a matrix, each row corresponds to each value in cross rule value  (previousrulevalue)    %so the number of rows in rulepercentage is equal to number of elements    %in previous rule value. each roq of rulepercentage adds to 100.    %second row of rule percentage indicates percentage distribution of people with property in cross rule col()    %cross rule col is another property which was distributed earlier(population in a region)         %rule2.rule=[1,2,3,4,5];  regions          %rule2.populate=[10,15,20,25,30];    percentage distribution is regions          %rule3.rule=[1,2,5]          educational level          %rule3.populate=          %          [20,30,50;          percentage distribution of people in region 1          %          25,25,50;          percentage distribution of people in region 2          %          10,40,50;          percentage distribution of people in region 3          %          33,33,34;          percentage distribution of people in region 4          %          75,20,5];          percentage distribution of people in region 5          %          %when we say cross rule, rule3 depends on rule2,          % it means that first fraction of rule2 (10%) has sub fractions, where as 20,30,50 % of them have different attributes          %similarly, people with second attribute 15% have sub fractions, where as 25%, 25%, 50% of themhave different attributes.          %this is like, say rule 2 applies to population in 5 regions          %rule 3 may apply to educational level.    [m,p]=size(rulepercentage);%dependent percentage values    [p,q]=size(previousrulevalue); %independent previous rule
   if m==q
      %valid cross rule
      ravi=[(1:length(inputcol))',previousrulecol,inputcol]; %ravi has index, independent column, and dependent column
      for i=1:length(previousrulevalue) %for al the independent rule values,
         b=ravi(find(ravi(:,2)==previousrulevalue(i)),:); %b has all rows  that match with independent rule in independent rule column
         b(:,3)=raghu_assignINDrule(b(:,3),rulevalue,rulepercentage(i,:));%b's dependent rule column is assigned rule based on the value as mentioned
         ravi(b(:,1),3)=b(:,3);%b is again inserted in to original ravi matrix in its own position.
      end
      result=ravi(:,3);
   else
      result=inputcol;
   End
```

```
function result=raghu_feeling(inputcol,v1,v2,type)
%can be considered as version 1.01
val1=v1;mean1=v1;val2=v2;var1=v2;AssignUnif=101;AssignNorm=102;ShiftUnif=201;ShiftNorm=202;
if(type<200) % assignment
   if type ==AssignUnif
         result=random('unif',val1,val2,length(inputcol),1);
   elseif  type == AssignNorm
         result=random('normal',mean1,var1,length(inputcol),1);
   end
```

```
elseif (type<300) %shift
   if type ==ShiftUnif
         result=inputcol+random('unif',val1,val2,length(inputcol),1);
   elseif  type == ShiftNorm
         result=inputcol+random('normal',mean1,var1,length(inputcol),1);
   end
else
   result=inputcol.*0;
end
```

```
function figID=raghu_hist(data,slots,color,bcolor,xmin,xmax,Plotmode,scalemode,xlab)
%can be consired as version 1.09
x123=xmin:(xmax-xmin)/(slots-1):xmax;a=histc(data,x123);
if(scalemode==1)
   a=a.*100/length(data);
end
figID=bar(x123,a,1);
if(Plotmode==1)
   ylim([0,length(data)]);
end
h=get(gca,'Children');set(h(1),'FaceColor',color,'EdgeColor',bcolor);hold off;grid on;xlabel(xlab);xlim([xmin xmax]);
```

```
function raghu_hist_redraw(data,slots,xmin,xmax,FID,scalemode)
%can be consired as version 2.01
x123=xmin:(xmax-xmin)/(slots-1):xmax;a=histc(data,x123);
if(scalemode==1)
   a=a.*100/length(data);
end
set(FID,'YDataSource','a');refreshdata(FID,'caller');
```

```
function result = raghu_SELECTIVEfeeling(inputcol,controlcol,controlval,controlmean,controlvar,type,xmin,xmax,EventTimeLine)
%can be considered as version 1.05
   if(EventTimeLine>0)
      if(EventTimeLine<1)
         controlmean=controlmean.*EventTimeLine;          controlvar=controlvar.*EventTimeLine;
      end
      if (unique(controlcol)==unique(controlval)')  %if all values from controlling column are covered and matching,
         a=length(controlval);          b=length(inputcol);  %number of control vars %number of elements in coltrol col
         ram=[(1:b)' inputcol controlcol];           rahim=[];
         for i=1:a%for all elements in control vals,
            hari=find(ram(:,3)==controlval(i)); ravan=ram(hari,:); %hari has all element indexes with control var i %ravan has rows with control var i
            if(controlvar(i)==0)
               %if variance is given as 0, it is assumed that we dont want that assignment to happen. this is useful while shiftingfeelings, instead of
shifting part bt part, we candirectly shift to certain location based on control val.
            else
               ravan(:,2)=raghu_feeling(ravan(:,2),controlmean(i),controlvar(i),type); % ravan now has assigned feelings
            end
            rahim=[rahim;ravan];%rahim has the matrix building up.
         end
         [gopi,harsha]=sort(rahim(:,1),'ascend'); %reconstruct based on oroginal indices
         result=rahim(harsha,2);          result(find(result<xmin))=xmin;          result(find(result>xmax))=xmax;
      else
         result=inputcol;
      end
   else
      result = inputcol;
   end
```

END MATLAB CODE