Hans-Gerlach Woudboer

The Zero-Sum Game in Activity Based Costing

When building an activity-based costing model the activities used and their data is most likely the task with the highest cost to perform properly. Also most likely the highest barrier to starting in the 1st place. Common methodologies are: measuring each activity, doing educated guesses with the department heads, doing time studies using various tools and taking available data, measuring the individual activities and how often the individual user has performed that. That could be any unit in time or it could be from reports collected within CRM software, like salesforce, etc.

Because of above necessary efforts, it's not a surprise that many activity-based costing projects have been abandoned or never been updated and becoming part of the business enhancement process as it could be.

When starting a new activity-based costing model, one of the methodologies is called a Rapid Prototyping Approach Of a Business Model or close to that notation-[this term is used by Gary Cokins and also by the author of this article].

However, due to the fact that there are data available for over thousand occupations and their related detailed work activities down to the task. Such a generic database-with about 202,000 details just on the median data-a similar amount is available for the upper and lower confidence interval. This generic database is used by RBM enabling a detailed business model targeting at the company in question, just by using a couple of accounting details combined with the detailed work activities of the people working in this enterprise. That way out of the zillions of possible business models, we are carving out the one which perfectly becomes the mirror image of the enterprise on your screen.

Let's not forget: "**Good Decisions Require Good Models**" so it's all about making the right decisions with confidence and as fast as possible and knowing what you're doing prior to execution.

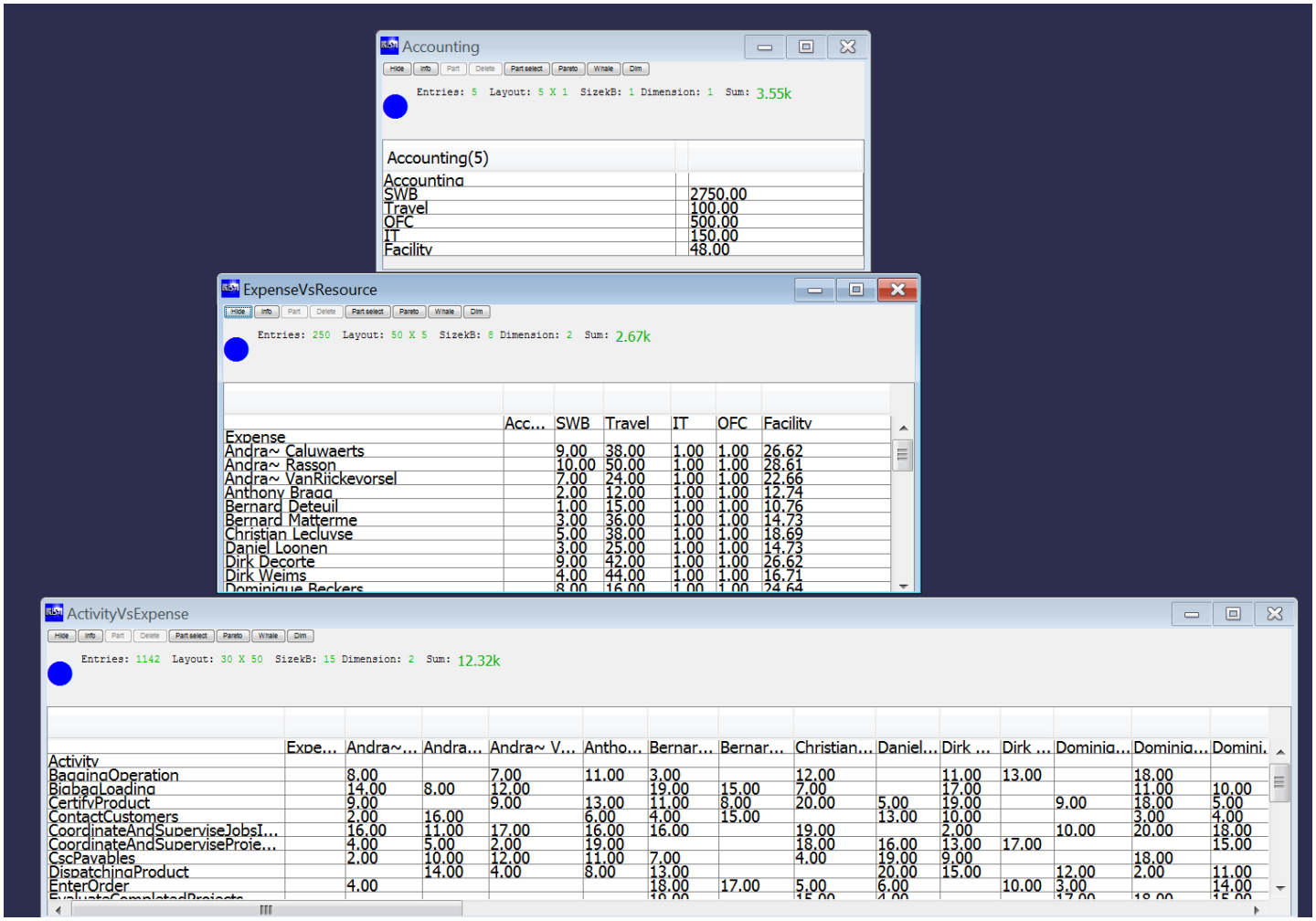
It enables you to calculate the P&L per customer, per order, by product or for any dimension interesting.

One of the questions to be examined: how good are those data of the generic detailed work activities versus the reality?

In this article we are carrying out an experiment based on a company with the following assumptions: we are in a B2B environment, with 50 people, producing 10 products selling it to 234 customers with just 30 work activities. Those data have been randomly generated.

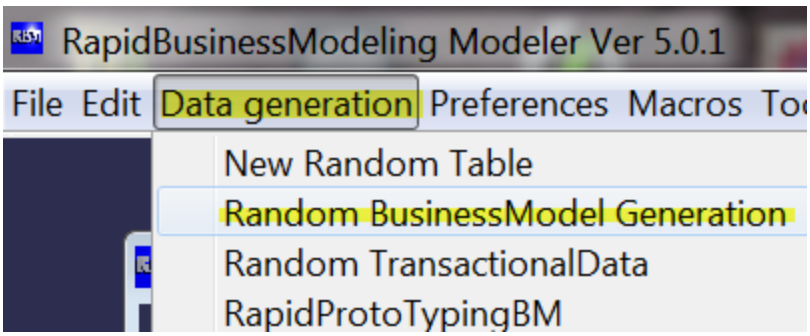
There are no energies or material or any other costs besides the cost of calculating the workplace cost of each person in the company.

Picture 1



Picture 1 shows 3 tables

By using the default of the Data generation function,



above tables Picture 1 among others are generated.

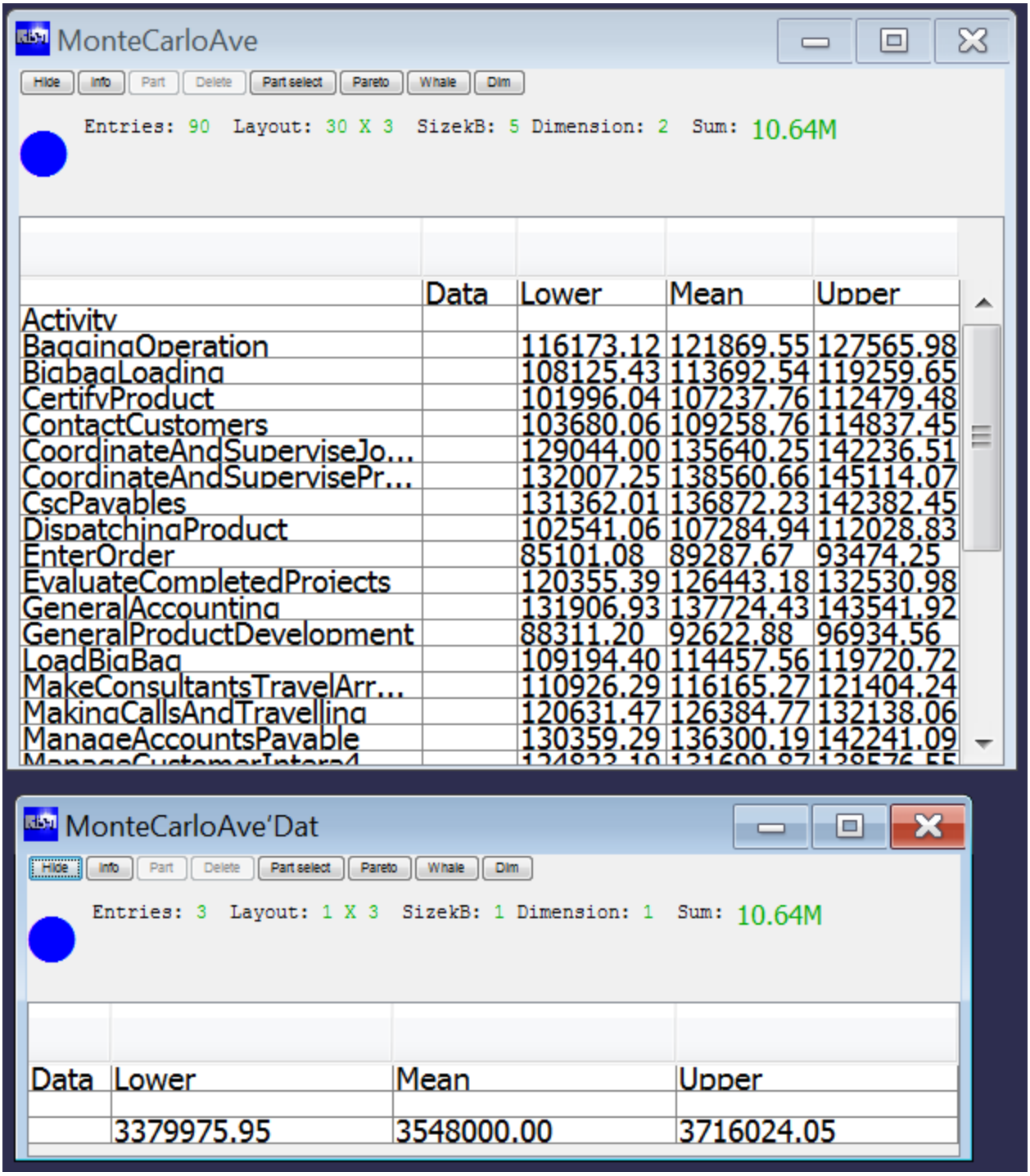
The table Accounting produced in millions was divided by thousand which is resulting in Accounting of 3.55K

Using a Smart Button Approach Makes It Real Easy

As a the next step we are simply pressing the smart button called **MonteCarlo Ave**which runs 100 decomposition calculations using a plus or -30% data variation of the table ActivityVsExpense for each decomposition step. Mathematica's RBM adapted uniform distribution function has been used.

After those hundred decomposition steps, we are using the statistics for mean, standard deviation and populating 3 activity tables: with the dimension Activity with a data field for Upper, Lower, Mean. The Monte Carlo simulation generates the following resulting tables.

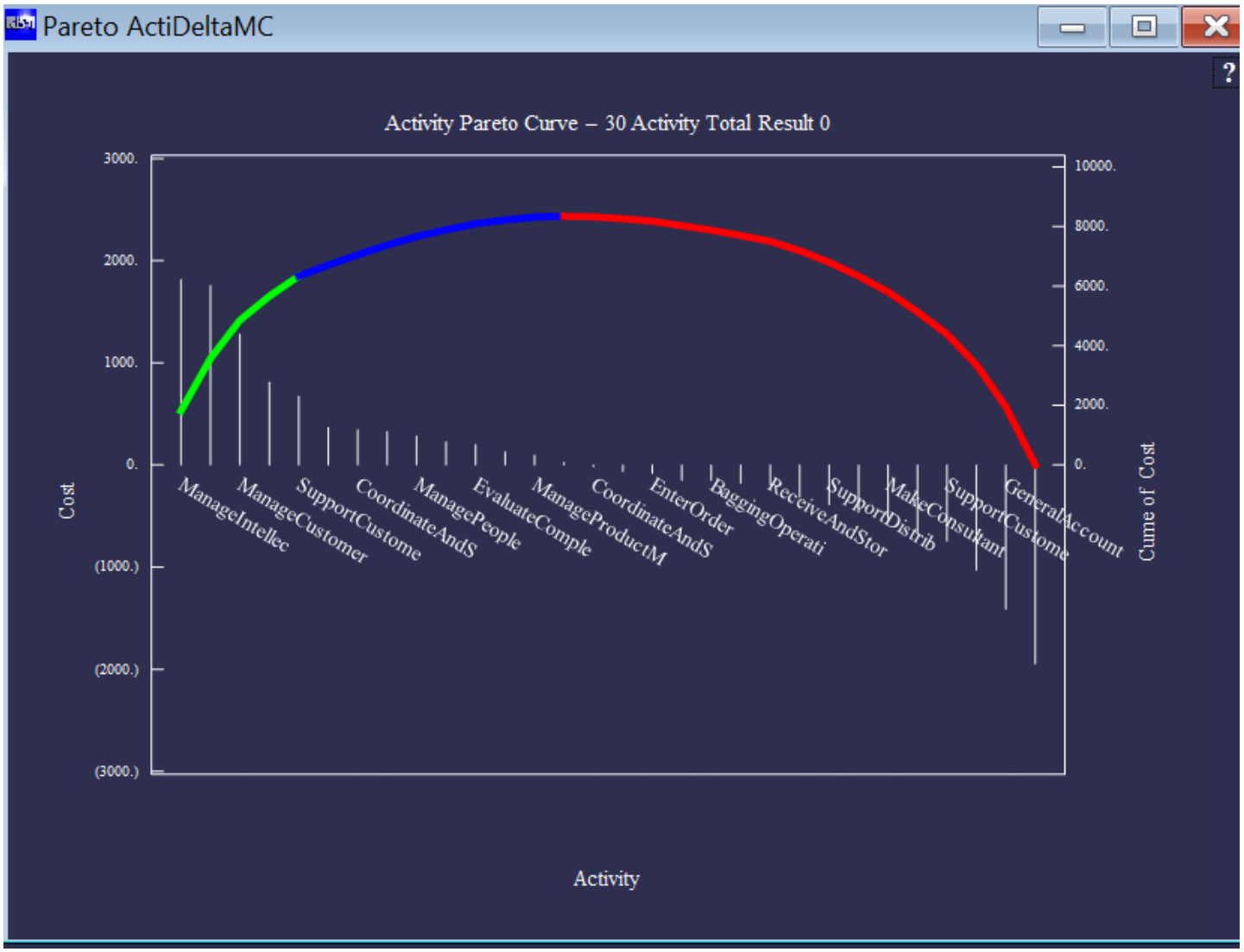
see more on Monte Carlo



Let's have a closer look to the upper confidence level. What does that mean? It means that there is a 5% probability that the costs are even higher than the upper confidence level shows.

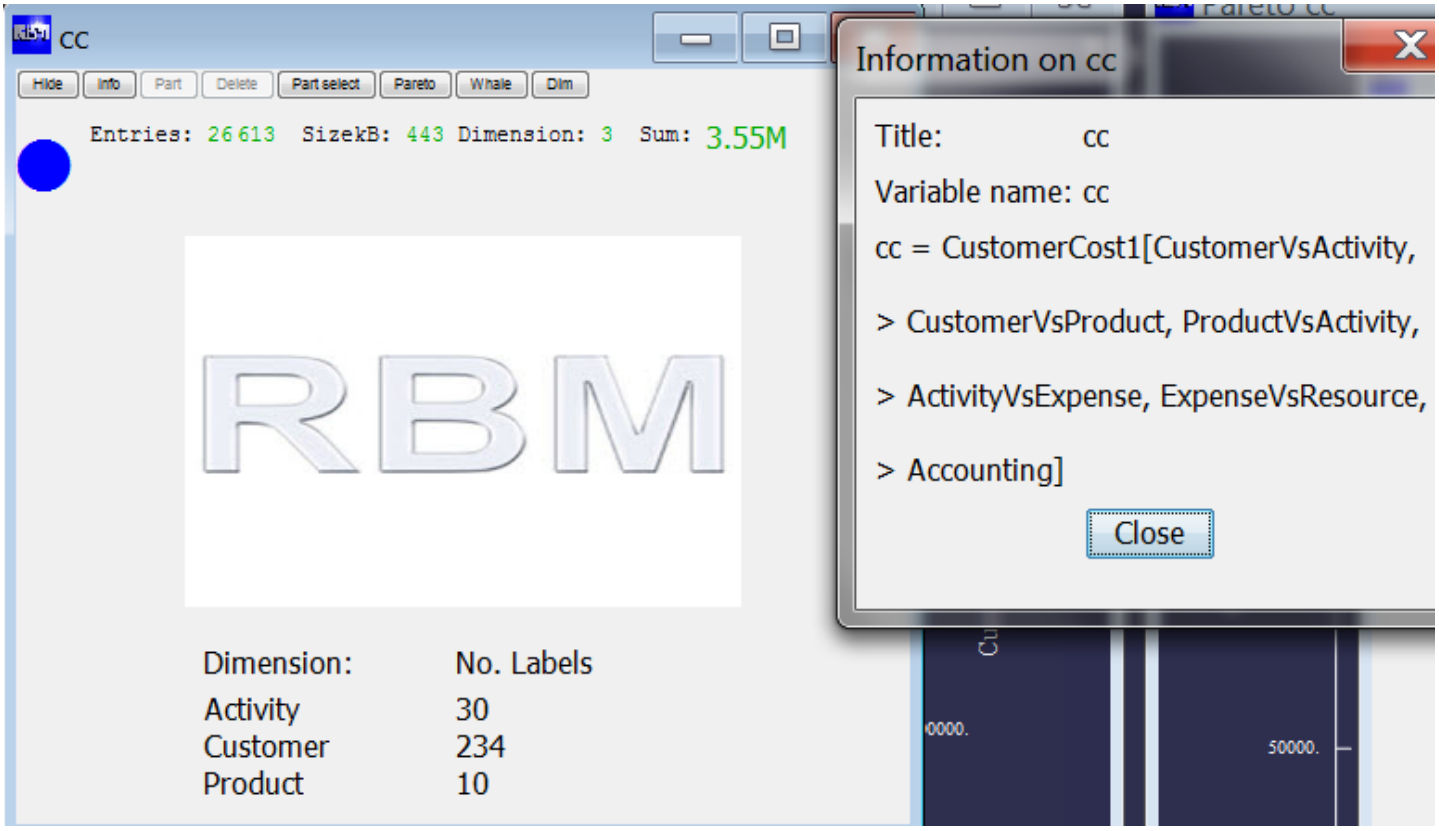
If you now decompose the resource cost of 3.55 million across the upper confidence level by taking a RBMDot Operator,we generate exactly the 3.55 m across all activities. And doing the same for the lower confidence level data, then we generate in both cases exactly the 3.55m. If you now **subtract** the resulting activity cost table produced by taking the upper confidence from the lower confidence table, then we are getting zero as a sum. Using a Pareto chart to display the resulting picture.

see Picture 3



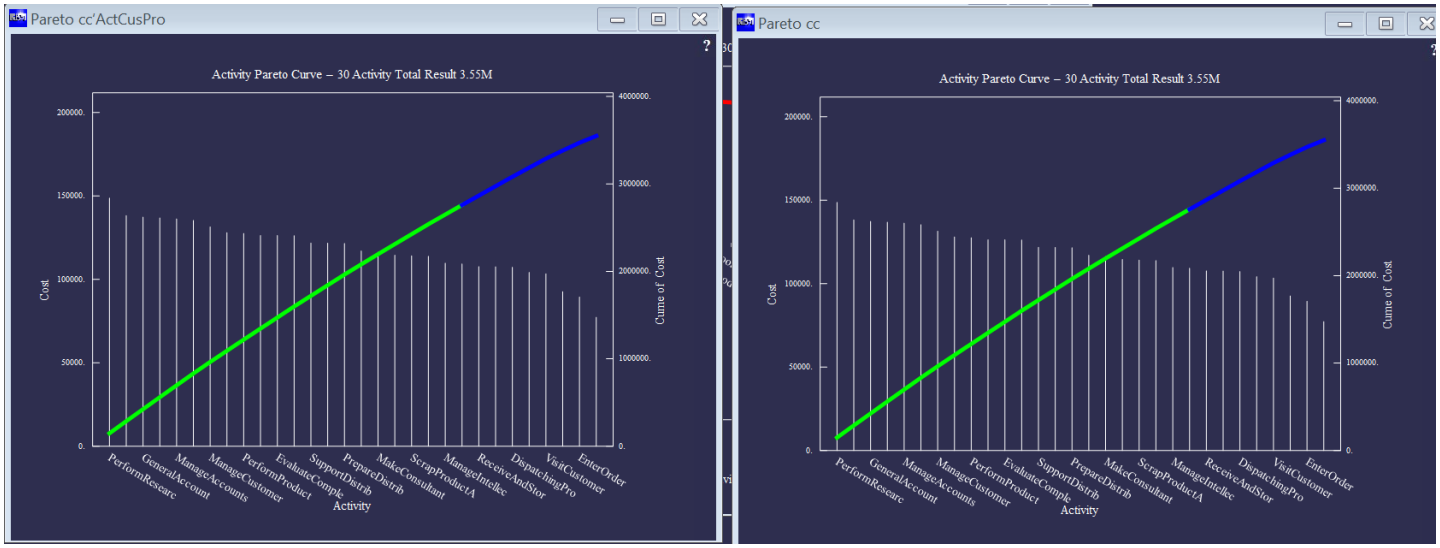
The calculation of the customer cost called **CC** by simply pressing the smart button on the Custom-Palette calculates automatically the customer cost **cc**

Pictures 3



The calculation process can be seen on the information on **CC.** It's simply the decomposition of **Accounting** via all those tables by using the function CustomerCost1.

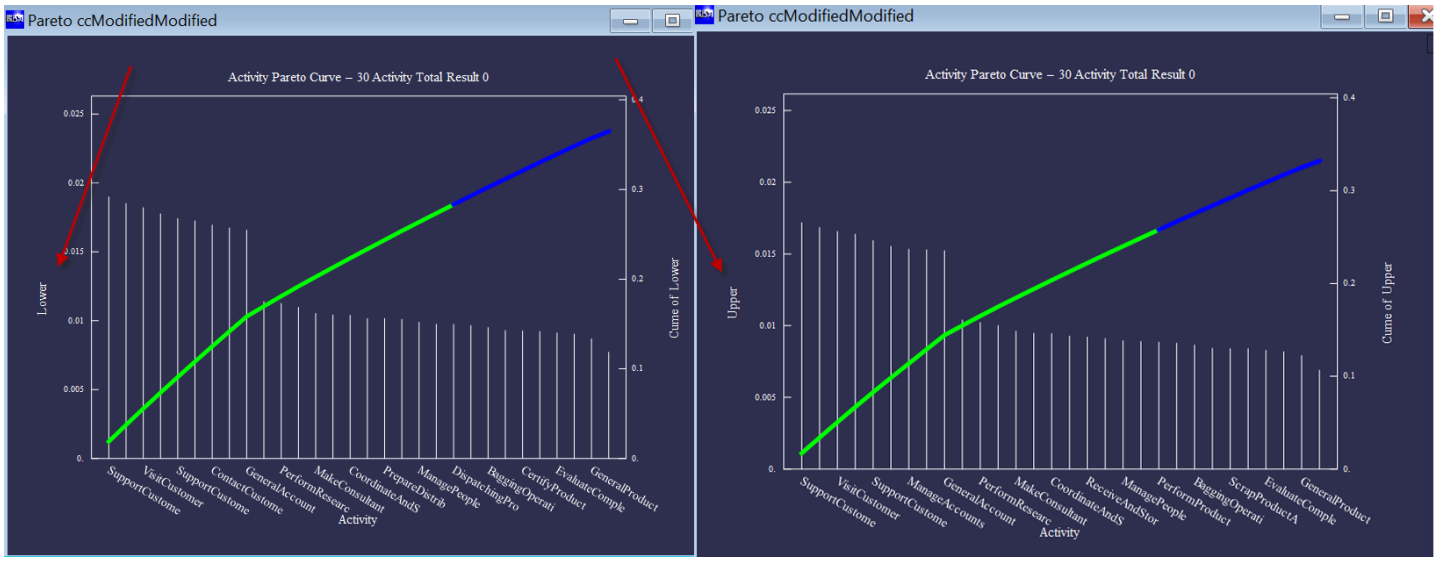
If you continue to decompose using the customer cost calculated from the mean and decompose it over the activity cost, it looks like the right part of the Pareto, whereas on the left side CC 'ActCusPro was calculated by decomposing CC via the driver upper minus lower cost:



Pareto displaying the activity cost for both processes, one cannot see any difference. Picture 4

Let's use another experiment and divide the table Monte Carlo with the upper and lower and mean by the sales table called CustomerVersusProduct and the display the upper and lower activity costs as a Pareto chart. That means we are looking at the unit costs for activities.

see Picture 5



Also interesting to see how part of this looks as a spreadsheet

mtmp.xls [Kompatibilitätsmodus]

	A	B	C	D	E	F	G
1	Customer	Product	Activity	Data			
2	PossessivesAG	Product10	BaggingOp	Lower		1.68E-05	
3	PossessivesAG	Product10	BigbagLoa	Lower		1.43E-05	
4	PossessivesAG	Product10	CertifyPro	Lower		9.85E-06	
5	PossessivesAG	Product10	Coordinate	Lower		8.57E-06	
6	PossessivesAG	Product10	Coordinate	Lower		2.35E-06	
7	PossessivesAG	Product10	EnterOrde	Lower		1.92E-05	
8	PossessivesAG	Product10	EvaluateC	Lower		3.45E-06	
9	PossessivesAG	Product10	LoadBigBa	Lower		1.83E-05	
10	PossessivesAG	Product10	MakeCons	Lower		1.56E-05	
11	PossessivesAG	Product10	MakingCal	Lower		9.85E-06	
12	PossessivesAG	Product10	ManageAc	Lower		3.04E-05	
13	PossessivesAG	Product10	ManageInt	Lower		1.45E-07	
14	PossessivesAG	Product10	ManagePr	Lower		7.47E-06	
15	PossessivesAG	Product10	PerformPr	Lower		1.57E-05	
16	PossessivesAG	Product10	PerformRe	Lower		1.22E-05	
17	PossessivesAG	Product10	PrepareDi	Lower		9.78E-06	
18	PossessivesAG	Product10	PriceNego	Lower		8.53E-06	
19	PossessivesAG	Product10	ReceiveAn	Lower		5.69E-06	
20	PossessivesAG	Product10	ScrapProd	Lower		1.2E-05	
21	PossessivesAG	Product10	SupportCu	Lower		1.67E-05	
22	PossessivesAG	Product10	SupportDi	Lower		1.39E-05	
23	PossessivesAG	Product11	BaggingOp	Lower		1.33E-05	
24	PossessivesAG	Product11	BigbagLoa	Lower		7.5E-06	
25	PossessivesAG	Product11	CertifyPro	Lower		1.6E-05	
26	PossessivesAG	Product11	Coordinate	Lower		1.47E-05	
27	PossessivesAG	Product11	Coordinate	Lower		1.68E-05	
28	PossessivesAG	Product11	Dispatchin	Lower		1.49E-05	
29	PossessivesAG	Product11	EnterOrde	Lower		1.92E-05	
30	PossessivesAG	Product11	EvaluateC	Lower		9.47E-06	

It obviously proves that the so-called Zero Sum Game often cited in Activity-based Costing holds true!

What are the consequences as a result of this?

Well in my opinion, it appears that taking a generic database like introduced, stemming from over thousand occupants and their related detailed work activities, properly clustered and named. When taking those for any organization, just picking their occupations and the numbers-how many of those.

Using this and carving out of the generic 202K entries large database, embracing all the related detailed work activities and their related tasks, then one gets according to above prove a properly decomposed activity cost. Further related/decomposed to their products and or services by using the related accounting data.

Particularly taking transactional data, Bill of material, production tables, all material and energy, depreciation and any other direct customer cost, one can map this all out into a generic costing, respective revenue map getting your detailed customer and product P&L costs available.

Of course those activities have to be validated and where necessary adapted. But those models show already a picture of any organization or company far beyond having interviews with the department heads to start with. In our opinion is better to take the rapid prototyping rout first and then validate using knowledgeable people of the organization as a 2nd step.

That is the way RapidBusinessModeling is dealing with a Detailed Business Model of an Enterprise. Furthermore it continues to utilize our two-step profit increase algorithm showing you promptly and easily the best rout forward. Any of the profit handles an organization has control over, can be used for simulating and playing with those profit handles to determine the best way forward.

When doing the proper change management and having the management supporting the overall approach-this is not and can never be a one-man show! Because we are talking about essential changes to the enterprise, but those which matter and not those where people with the loudest voice getting too much attention. Provided your management is convinced and fully supporting the project, one can expect to enhance the overall profitability by around 30% within months.

RapidBusinessModeling's Technology & Solution

Turn's words into data and then make those words becoming factual helping your company to prosper and adapt their strategy in relation to their new insights. Besides that, you are now **working on your enterprise** versus going the continuous improvement rout taking forever!

Not that it's bad, but you are losing track and you might even change your job and it all boils down to Einsteins anecdote "***People love shopping wood, because they can immediately see their success!***" RBM enables you to go that way easily, just have a try at our AWS instance and take your own model in order to get the feeling providing you with the trust and confidence.

And of course RBM is more than happy to do it for you. Why not schedule a demonstration for your company see http://RapidBusinessModeling.com

Applying Monte Carlo simulation on the Activity Versus Expense matrix to be used for the decomposition of resource costs via the workplace cost into activity costs.

This example is simplified and is using the 11 activities carried out by 10 people in a small organization with 710 K total cost.

Label Title:	Dimension	3028 Bytes	83 Entries
Activity	11		
Expense	10		

Just have a closer look at 2 Activities where the table name is called **the function** name **Take2** just takes the 1st 2 elements see picture below

Take2[ActivityVsExpense]

Activity	Expense	
AssignConsultantsAndDevelopSchedulesForProjects	Bernard Dieu	12
AssignConsultantsAndDevelopSchedulesForProjects	Carl Haspeslagh	9
AssignConsultantsAndDevelopSchedulesForProjects	Frederic DeTry	5
AssignConsultantsAndDevelopSchedulesForProjects	Godelieve DeMoor	2
AssignConsultantsAndDevelopSchedulesForProjects	Hubert Haspeslagh	3
AssignConsultantsAndDevelopSchedulesForProjects	Nigel Brothearts	7
AssignConsultantsAndDevelopSchedulesForProjects	Paul Vercruysse	18
AssignConsultantsAndDevelopSchedulesForProjects	Ronald Goovaerts	20
AssignConsultantsAndDevelopSchedulesForProjects	Rony Devresse	4
EnterOrder	Bernard Dieu	5
EnterOrder	Frederic DeTry	8
EnterOrder	Godelieve DeMoor	5
EnterOrder	Hubert Haspeslagh	18
EnterOrder	Nigel Brothearts	1
EnterOrder	Ronald Goovaerts	7

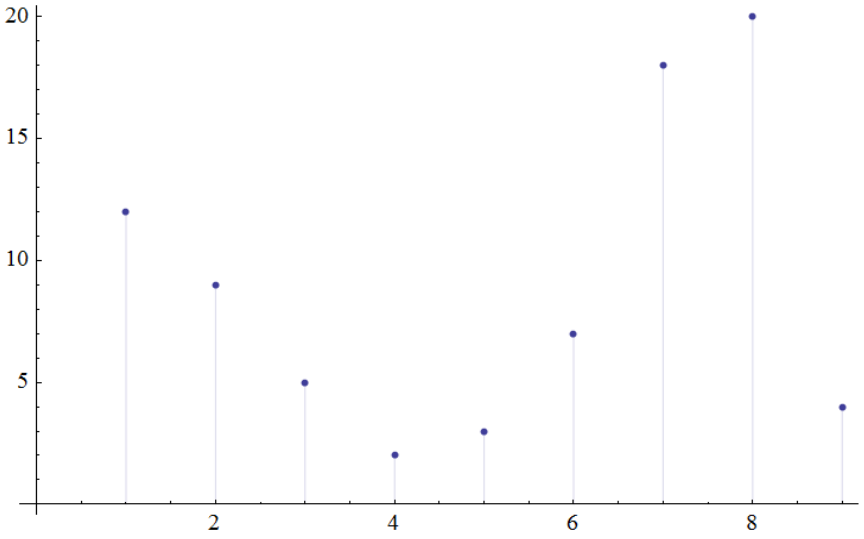
A screenshot with 10 people and their 11 Activities -some are truncated

Activity / Expense	AssignConsultantsAndDevelopSchedulesForProjects	EnterOrder	MakingCallsAndTravelling	ManageBusiness	ManageProductMix2	OtherSupportActivities	PaManagement	PrepareProjectInvoices	RepresentBusiness	ShutdownLine	S
Bernard Dieu	12	5	11	4	11	1	8	15	8	□	
Carl Haspeslagh	9	□	8	3	4	17	14	6	5	4	
Frederic DeTry	5	8	18	6	7	18	13	9	□	□	
Godelieve DeMoor	2	5	□	9	12	6	□	□	□	□	
Hubert Haspeslagh	3	18	14	□	20	6	4	□	9	□	
Nigel Brothaerts	7	1	□	7	7	3	12	6	3	16	
Paul Vercruysse	18	□	□	7	6	17	□	13	12	16	
Ronald Goovaerts	20	7	20	13	15	11	□	7	19	□	
Rony Devresse	4	□	2	15	□	14	1	10	15	4	
Sam Schoondetwoert	□	□	19	18	12	□	20	□	19	2	

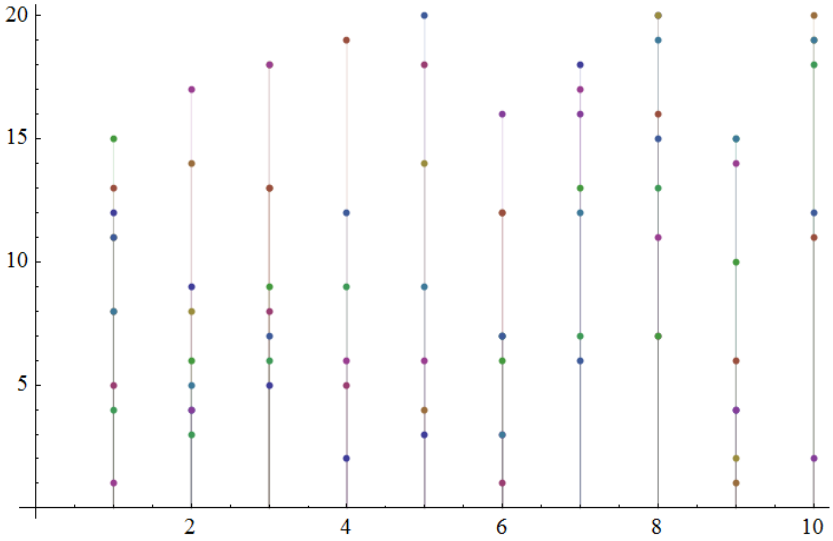
In the next picture one can see the plot of the data without any dimension and labels. The picture shows the 1st activity for 10 expenses

1 Activity, 10 Expense aka People

```
ListPlot[Normal[Take2[ave, 1]], Filling -> Axis]
```



```
ListPlot[Normal[Take2[ave, 11]], Filling -> Axis]
```



The next plot shows 11 Activity and 10 Expense aka People

Below is a function **USD** which stands for UniformSumDistribution. When applying this function, a random value between .7 and 1.3 is taken to multiply the **ActivityVsExpense** aka **ave** table.

```
usd[x_]:=Random[UniformSumDistribution[2,0.7` x,1.3` x]]
```

In the picture below one can see 2 results of using above function. The function Range[2] does it 2 times.

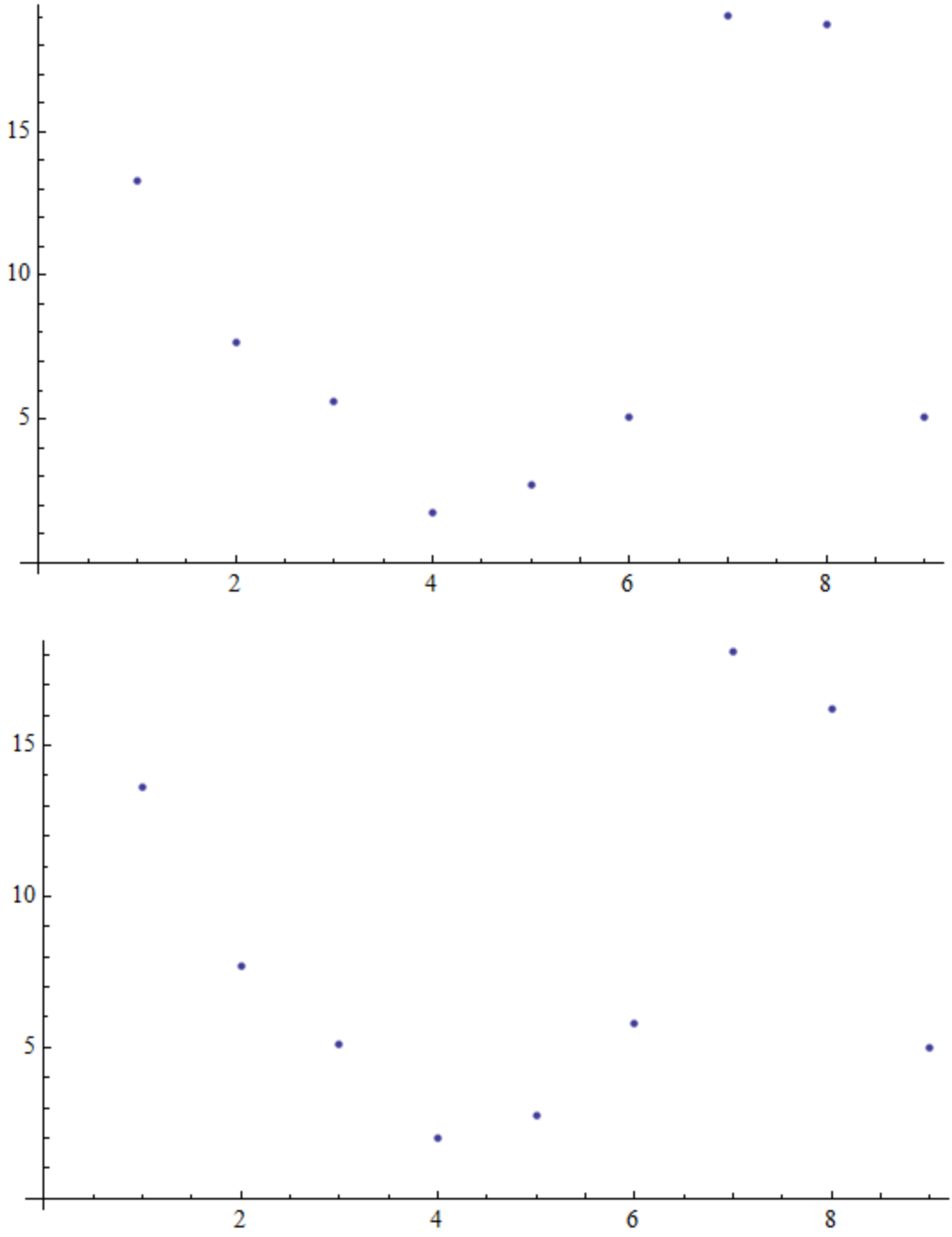
```
Map[usd[Take2[ave]] &, Range[2]]
```

Activity	Expense	
AssignConsultantsAndDevelopSchedulesForProjects	Bernard Dieu	10.9437
AssignConsultantsAndDevelopSchedulesForProjects	Carl Haspeslagh	9.32118
AssignConsultantsAndDevelopSchedulesForProjects	Frederic DeTry	5.78487
AssignConsultantsAndDevelopSchedulesForProjects	Godelieve DeMoor	2.42442
AssignConsultantsAndDevelopSchedulesForProjects	Hubert Haspeslagh	2.81073
AssignConsultantsAndDevelopSchedulesForProjects	Nigel Brothearts	7.94071
AssignConsultantsAndDevelopSchedulesForProjects	Paul Vercruysse	16.593
AssignConsultantsAndDevelopSchedulesForProjects	Ronald Goovaerts	16.5837
AssignConsultantsAndDevelopSchedulesForProjects	Rony Devresse	4.69806
EnterOrder	Bernard Dieu	5.35897
EnterOrder	Frederic DeTry	6.905
EnterOrder	Godelieve DeMoor	5.244
EnterOrder	Hubert Haspeslagh	17.1207
EnterOrder	Nigel Brothearts	1.01311
EnterOrder	Ronald Goovaerts	7.65923

Activity	Expense	
AssignConsultantsAndDevelopSchedulesForProjects	Bernard Dieu	13.8872
AssignConsultantsAndDevelopSchedulesForProjects	Carl Haspeslagh	8.69864
AssignConsultantsAndDevelopSchedulesForProjects	Frederic DeTry	6.17715
AssignConsultantsAndDevelopSchedulesForProjects	Godelieve DeMoor	1.67466
AssignConsultantsAndDevelopSchedulesForProjects	Hubert Haspeslagh	3.18274
AssignConsultantsAndDevelopSchedulesForProjects	Nigel Brothearts	7.45877
AssignConsultantsAndDevelopSchedulesForProjects	Paul Vercruysse	21.5602
AssignConsultantsAndDevelopSchedulesForProjects	Ronald Goovaerts	20.3348
AssignConsultantsAndDevelopSchedulesForProjects	Rony Devresse	3.78656
EnterOrder	Bernard Dieu	5.01813
EnterOrder	Frederic DeTry	9.21971
EnterOrder	Godelieve DeMoor	6.01362
EnterOrder	Hubert Haspeslagh	23.3136
EnterOrder	Nigel Brothearts	0.936209
EnterOrder	Ronald Goovaerts	5.31838

When stripping off dimensions Activity and Expense and their respective labels. One can see just the numbers of the array. And in the picture below. It's just for the 1st activity. While the USD function is applied just once and when execute again the 2nd figure is generated.

```
ListPlot[Normal[usd[Take2[ave,1]]]]
```



Mathematica is working with lists and lists of lists and the below example just 2 activities. One time mapped.

```
Map[Normal[usd[Take2[ave]]] &, Range[1]]
```

```
{{{11.8082, 8.87263, 4.84388, 1.69175, 3.30378, 7.52566, 18.3011, 17.2581, 4.08931, ⬚},
  {4.34151, ⬚, 8.56249, 4.14906, 19.5533, 0.843965, ⬚, 7.82261, ⬚, ⬚}}}
```

```
Normal[Take2[ave,4]]
```

$$\begin{pmatrix} 12 & 9 & 5 & 2 & 3 & 7 & 18 & 20 & 4 & \square \\ 5 & \square & 8 & 5 & 18 & 1 & \square & 7 & \square & \square \\ 11 & 8 & 18 & \square & 14 & \square & \square & 20 & 2 & 19 \\ 4 & 3 & 6 & 9 & \square & 7 & 7 & 13 & 15 & 18 \end{pmatrix}$$

And just to show how 4 activities in matrix form look like.

Now let's apply some of the statistics on to the activity versus expense table just for 2 activities. So here we calculated the Mean, the Standard Deviation. Those are the components from which we calculated the upper, lower and mean values for the activity versus expense table. Mean plus or -2 Sigma stands for plus or -95%.

With other words, there is a 5% probability that the values are even higher or lower than the upper and lower boundaries suggests.

```
mean = Mean[Map[Normal[usd[Take2[ave]]] &, Range[100]]]
```

```
{{12.0121, 9.08862, 5.03888, 1.99305, 2.9728, 6.83056, 17.7431, 19.6956, 4.02964, □},
 {5.00103, □, 7.93612, 5.03044, 18.1081, 0.996242, □, 6.9246, □, □}}
```

```
sigma = StandardDeviation[Map[Normal[usd[Take2[ave]]] &, Range[100]]]
```

```
{{1.61756, 1.05009, 0.575358, 0.286174, 0.392598, 0.789221, 2.14652, 2.54891, 0.513699, 0.},
 {0.660371, 0., 1.05731, 0.563416, 2.31157, 0.111133, 0., 0.828454, 0., 0.}}
```

```
upper = mean + 2 sigma
```

```
{{15.2472, 11.1888, 6.1896, 2.56539, 3.758, 8.409, 22.0361, 24.7934, 5.05703, 0.},
 {6.32177, 0., 10.0507, 6.15728, 22.7312, 1.21851, 0., 8.58151, 0., 0.}}
```

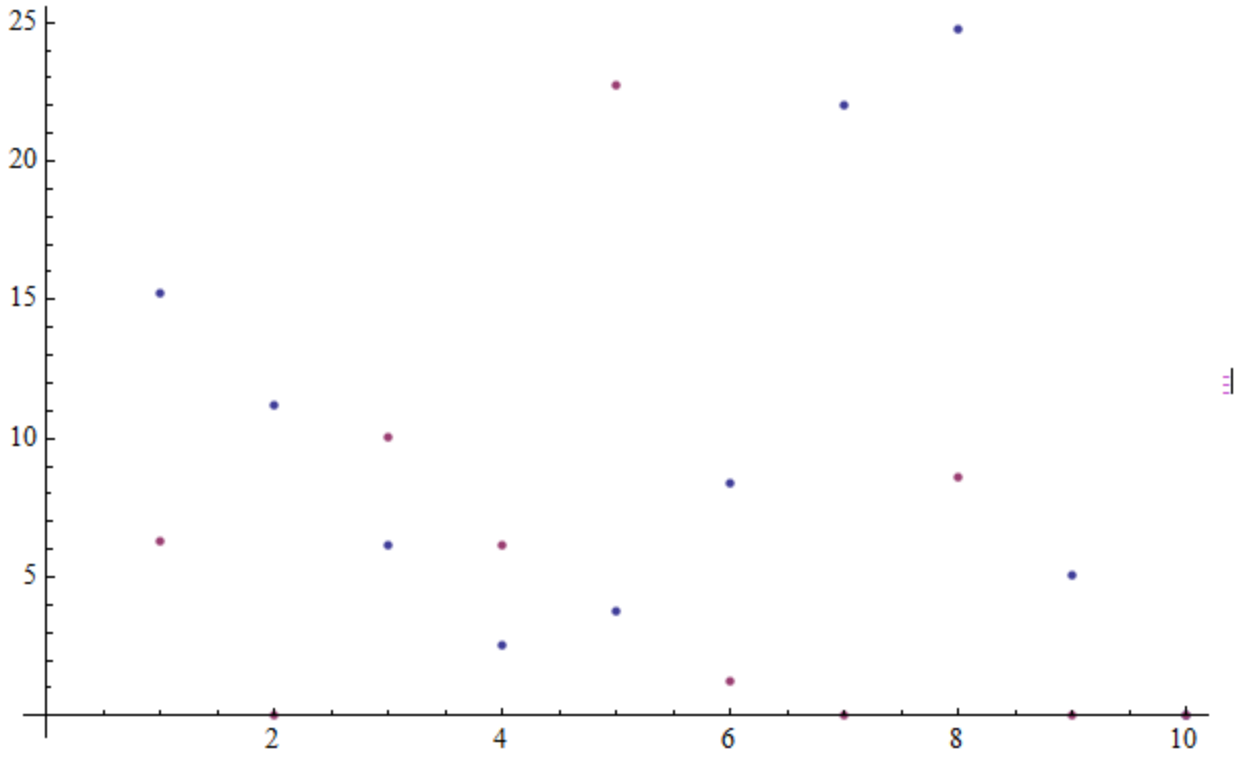
```
lower = mean - 2 sigma
```

```
{{8.77699, 6.98844, 3.88817, 1.4207, 2.18761, 5.25212, 13.4501, 14.5978, 3.00224, 0.},
 {3.68029, 0., 5.82149, 3.90361, 13.485, 0.773976, 0., 5.26769, 0., 0.}}
```

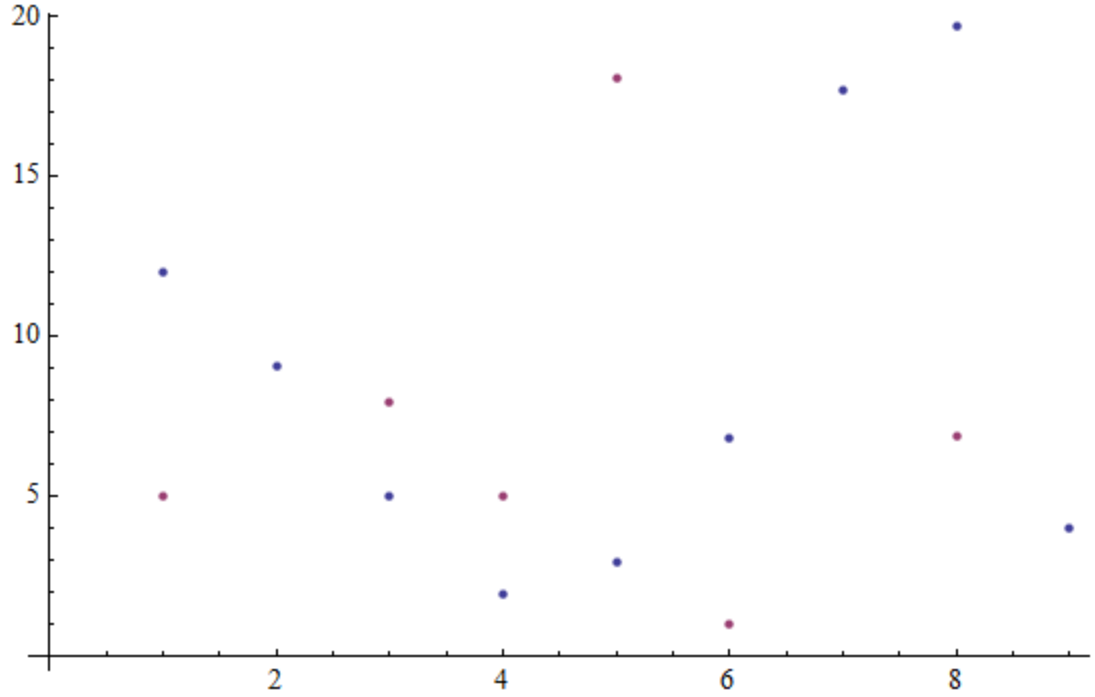
Below are some figures showing 2 activities and their related expense of the upper, mean, lower and a sigma value

t Take2[ave]

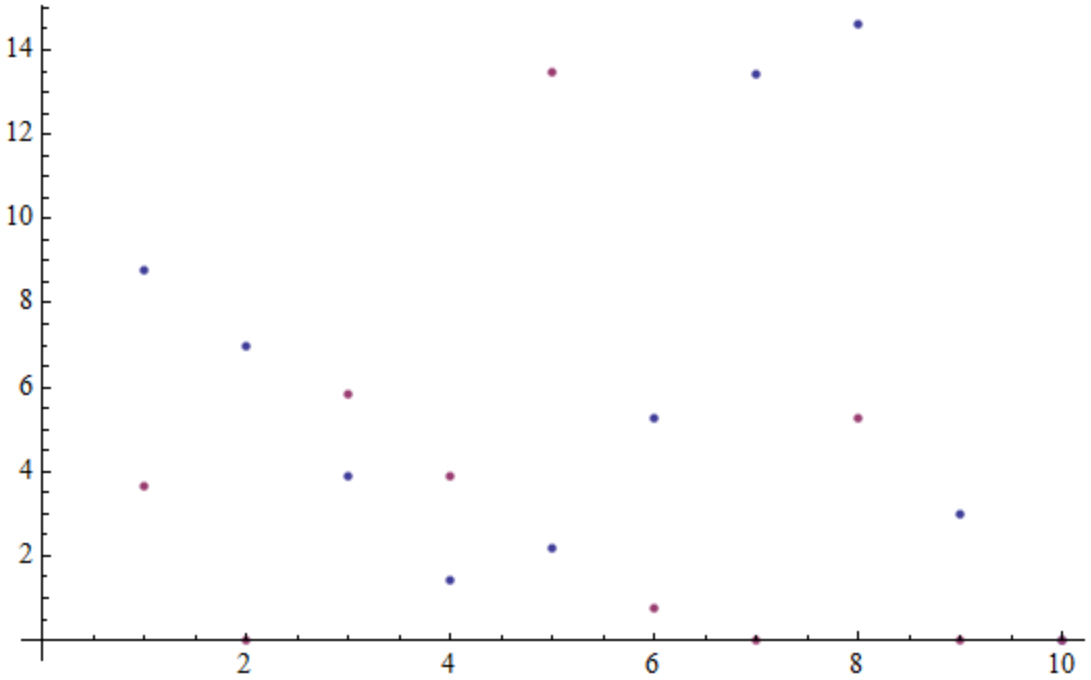
```
ListPlot[upper]
```



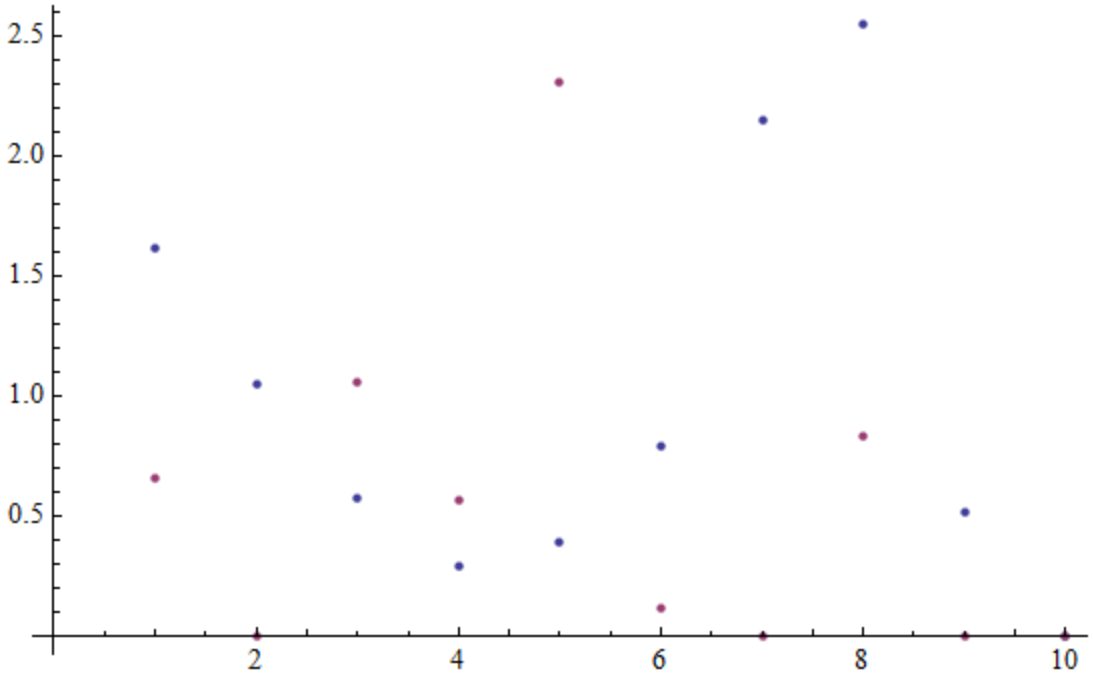
```
ListPlot[mean]
```




```
ListPlot[lower]
```



```
ListPlot[sigma]
```



Basics of statistics

Given a list with n elements x_i, the *mean* `Mean[list]` is defined to be

$\mu(x) = \bar{x} = \sum x_i / n.$

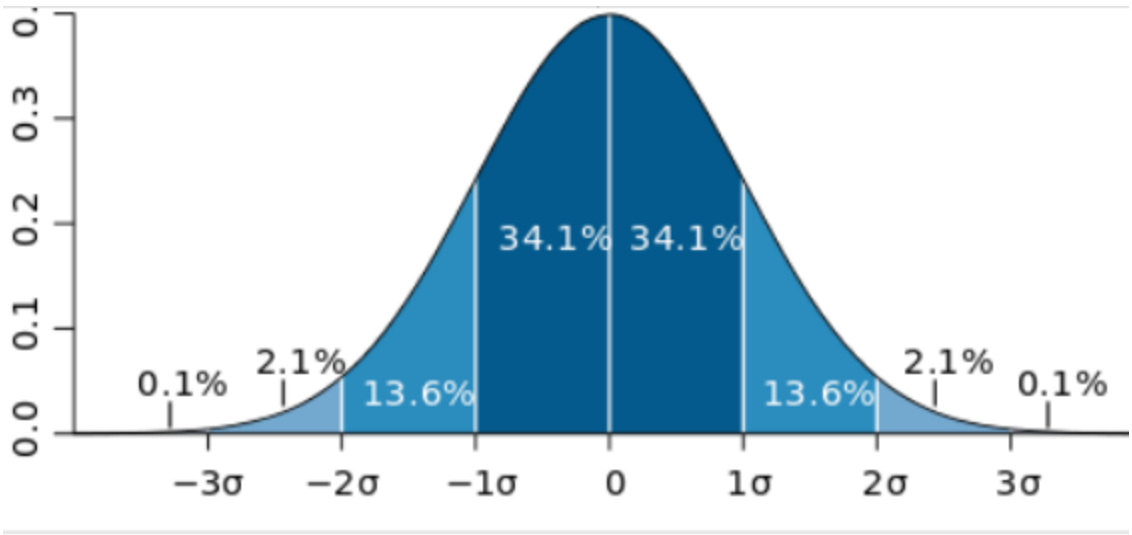
The *variance* `Variance[list]` is defined to be

$\text{var}(x) = \sigma^2(x) = \sum (x_i - \mu(x))^2 / (n-1)$, for real data. (For complex data

$\text{var}(x) = \sigma^2(x) = \sum (x_i - \mu(x)) (\overline{x_i - \mu(x)}) / (n-1).$)

The *standard deviation* `StandardDeviation[list]` is defined to be

$\sigma(x) = \sqrt{\text{var}(x)}$.



In statistics, the **standard deviation** is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean of the set, while a high standard deviation indicates that the data points are spread out over

ConfidenceIntervalDefinition