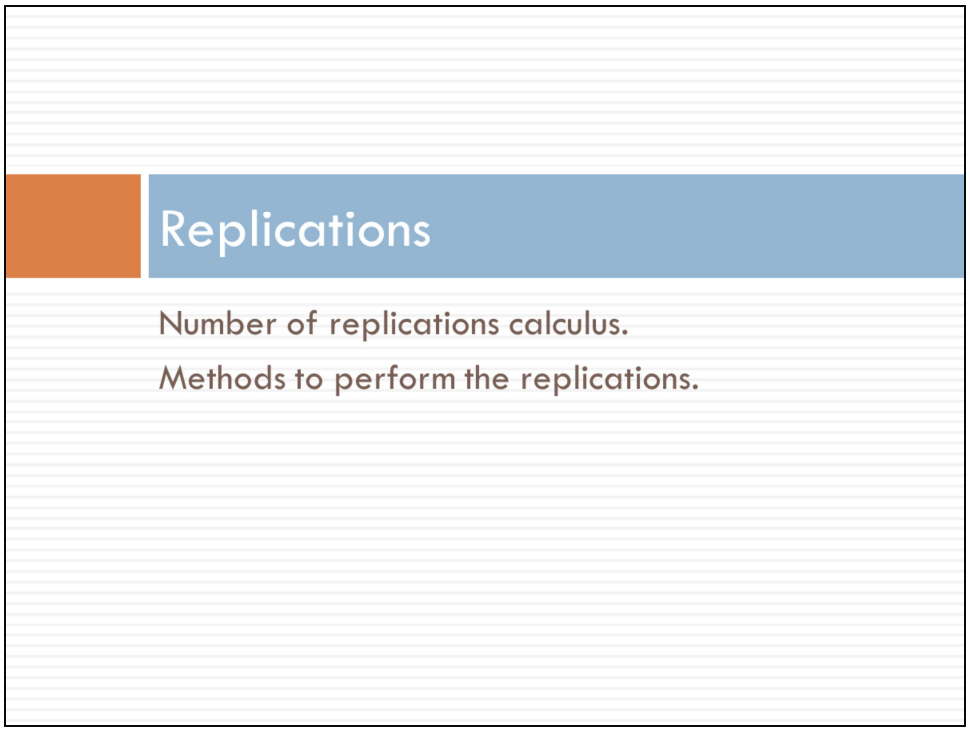
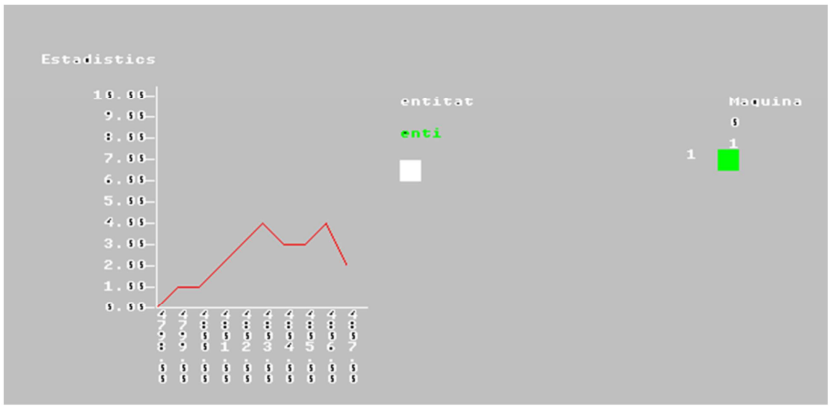


Now we are going to discuss how to deal with replications, first understanding if the number of replications are enough, and secondly, analyzing how to perform this replications and trying to reduce the number to execute.

Interest variable calculus



First, loot at this slide. For any interest variable the temporal chart we will obtain during the execution of the simulation model, will be like this. Think that this time Serie represents the mean number of elements in a queue. At the beginning of the simulation this value will be zero, but at the end of the simulation, and if the system is stable, we will see the the chart stabilizes around an specific value.

Experimentation

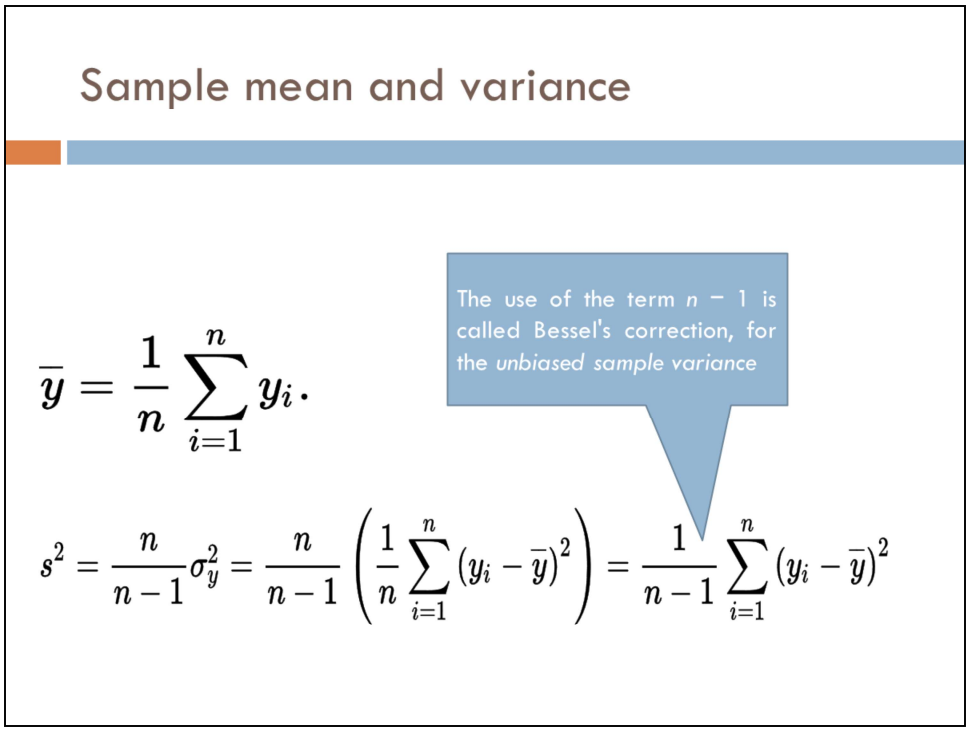
- Be x an interest variable

  x11,…x1i,…,x1m
  x21,…x2i,…,x2m
  ……
  xn1,…xni,…,xnm

- *n* is the number of replications.
- *xi* is the value of each one of the replications.

To recover the different observations we have of the interest variable, maned X, we will define an structure like the one presented here.

Each row is a set of observations that defines the time series for the replication. In this slide you can see that we have n replications and m observations for each replication.



Notice that we use the term n-1 to calculate the variance because we want to calculate an unbiased sample variance.

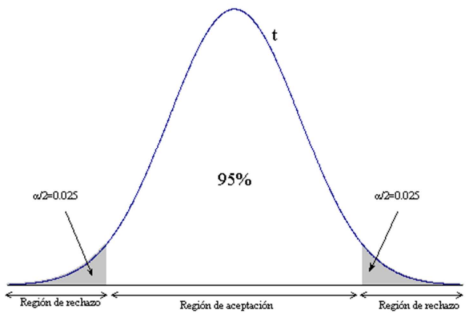
Confidence interval

- Need to know how far is μ and $\bar{y}$.
- Student's t-distribution of n-1 degrees of freedom.

$$\bar{y} \pm t_{1-\alpha/2,n-1}\sqrt{\frac{S^2}{n}}$$

With the sample mean and variance, we can calculate the confidence interval for the interest variable using this well-known formula.

Student's t-distribution



Notice that we will use a t-Student distribution, with two-tails, as is shown in this slide.

What is the correct n?

Replication	Value from the model
1	28.841
2	35.965
3	31.219
4	37.090
5	38.734
6	30.923
7	30.443
8	32.175
9	30.683
10	28.745

For the example, think that we have a simulation model, and we execute a scenario of this simulation model 10 times. We choose 10 times because is a common number to perform initially a scenario. If the experiment is not based in simulation, think in a chemical process or similar, maybe 10 is too much to start the analysis and we can do only 3 replications for each scenario. However in computing, starting with 10 is quiet common. In brief, the number varies from 3 to a higher value, depending on how expensive is to execute each one of the replications.

Slide 77

PFiC1 Cal recalcular

Pau Fonseca i Casas; 10/6/2019

Calculus of S an X

$$\overline{y} = 32.4818$$

$$S^2 = 3.5149$$

With this values we will obtain this mean and variance from the sample.

Calculus of the self-confidence interval

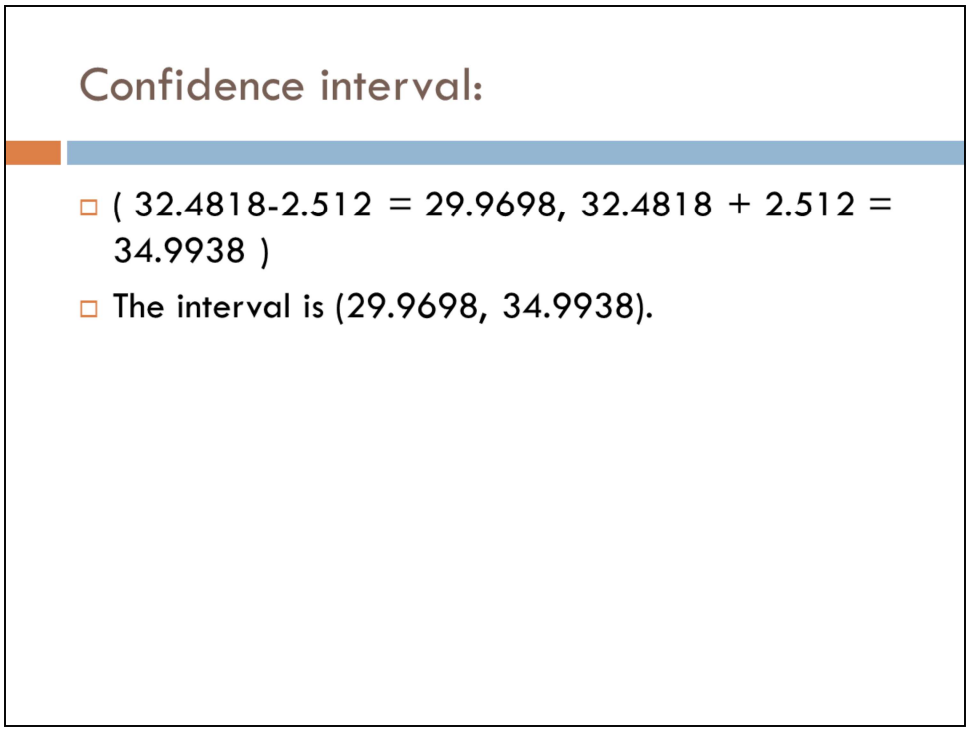
$$h = t_{1-\alpha/2,n-1}\frac{S}{\sqrt{n}}$$

$t_{9,0.975} = 2,26$

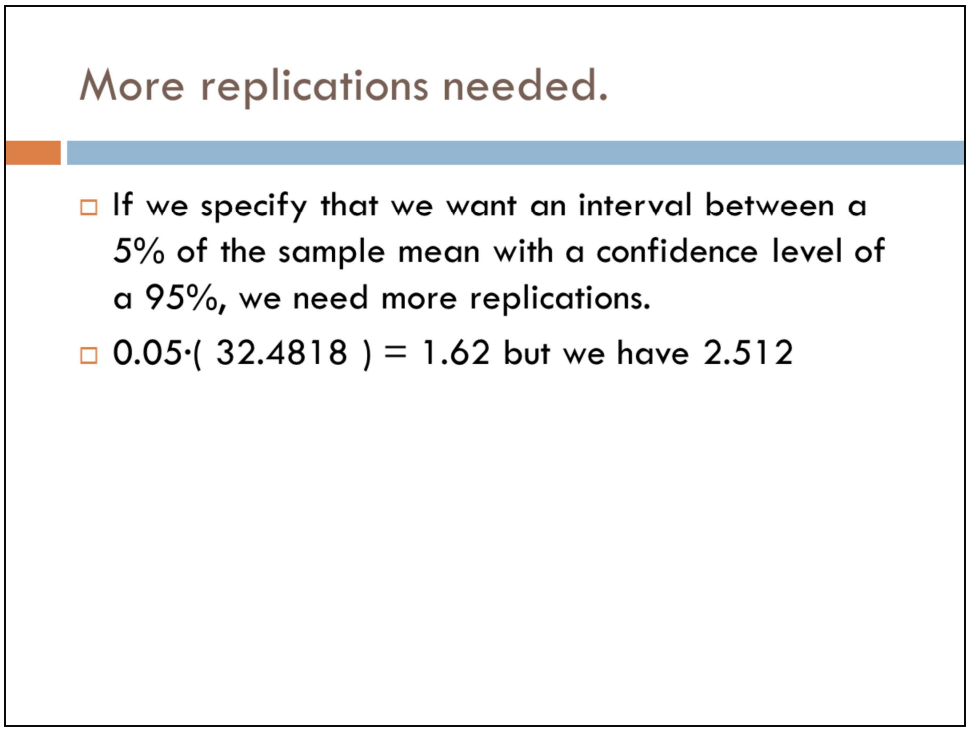
$h = 2,512$

	1 – a							
r	0.75	0.80	0.85	0.90	0.95	0.975	0.99	0.995
1	1.000	1.376	1.963	3.078	6.314	12.706	31.821	63.657
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355
9	0.703	0.883	1.100	1.383	1.833	**2.262**	2.821	3.250
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169

And then we can calculate the confidence interval for the value answer we are analyzing. Remember that for calculate the value of the t distribution you can always use the tables, considering the degrees of freedom you have, in that case nine, since 10 minus one, and considering that we are working with two tails, hence 0.975. On red is the value we must use in this case.



With this values we can calculate the confidence interval. The interpretation is that with a probability of 0.95, the random interval (29.96 to 34.99) includes the real value of the mean.



We need more replications if this interval is not enough to make decisions., think that you want more precision on the model because a deviation bigger that 5% implies to lose, or to win, some money.

In that case we must specify how big we will tolerate the confidence interval, referring to the value of the answer we obtain. Suppose that we desire no more that 5% for this confidence interval, calculating what this means imply to add the value that we obtain for the mean with a 5% to obtain the upper bound, and rest the 5% of the mean value to obtain the lower bound. To calculate the value of the interval is just to multiply the mean by 5%. This value, in that case is 1.62. Notice that this value is small that the 2.51 value that we obtain in the process we do in the previous slides. This implies that we need more replications.

Number of needed replications

- where:
- n = initial number of replications.
- n^* = total replications needed.
- h = half-range of the confidence interval for the initial number of replications.
- h^* = half-range of the confidence interval for all the replications (the desired half-range).

$$n^* = n\left(\frac{h}{h^*}\right)^2$$

The number of replications that we need can be calculated using the synthetic formula. On it we define the number of initial replication, value n, the half range of the confidence obtained with this initial number of replications, value h, calculated from the 10 replications presented on the previous table. We also need the h star, that represents the half-range of the confidence interval desired, in that case the 1.62 calculated on the previous slide.

With this we will obtain the n star, that represent the number of needed replications.

Number of replications calculus.

$$n^* = 10\left(\frac{2.512}{1.62}\right)^2 = 24.04$$

In this case, the n star is 24.04. Since we cannot do a half of a replication, this implies that we must do 25 replications. Since we already have 10, we continue the execution of the simulation model scenario with 15 more replications. Remember that the only element that we change is the random number stream used to generate the values of the statistical distributions used.

More replications…

Replication	Value from the model
11	33.020
12	29.472
13	27.693
14	31.803
15	30.604
16	33.227
17	28.085
18	35.910
19	30.729
20	30.844
21	32.420
22	39.040
23	32.341
24	34.310
25	28.418

On this table are the results we obtain from the execution of 15 more replications, and with this values we will be able to calculate a new mean and a new variance.

New mean and variance

$$\overline{y} = 32.1094$$

$$S^2 = 3.1903$$

The values that we are obtaining now are presented on this slide, they are going to be good, depending on the confidence interval they will generate.

New self-confidence interval

- In that case is enough, but the process can be iterative.

$$h = t_{1-\alpha/2,n-1}\frac{S}{\sqrt{n}}$$

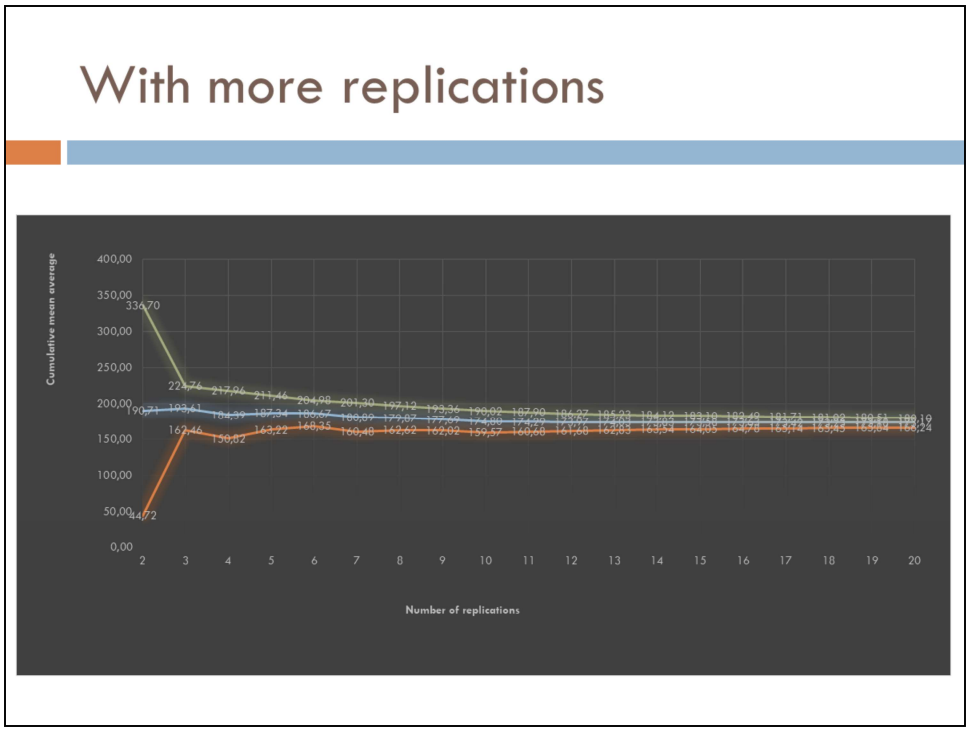
h = 1.3144 < 1.62

$$n^* = n\left(\frac{h}{h^*}\right)^2$$

The half range of the confidence interval, in that case, is 1.3, that is smaller that 1.605 (now we must calculate again the desired half-range because the values changes), and if we calculate the n star, we will see that now the value is about 16. This means that the number of replications are now enough.

As you will notice, now it seems that the number of replications that we need are not as bigger as 25, and maybe wit less replications will be enough. This can be true but depends on the ability of the modeler to define correctly the different replications, to copse with the uncertainly and recover as maximum variability of the answer with less replications. This is the reason that some commercial tools continuously calculate the number of replications needed, and automatically adds new replications until this condition is meet.

But be aware, this calculus is an iterative process, and the opposite can happen, that the number of replications growths. This is quite usual if the system is not stable and is not bounded by time, we follow an approach named Batch Means that we will see lather.



Finally, just to show how this iterative process looks like, as you can see, with two replications, the confidence interval obtained is huge, and it seems that a lots of replications are needed. Adding more replications makes the confidence interval small and small. Did you notice that it seems to converge at infinity?

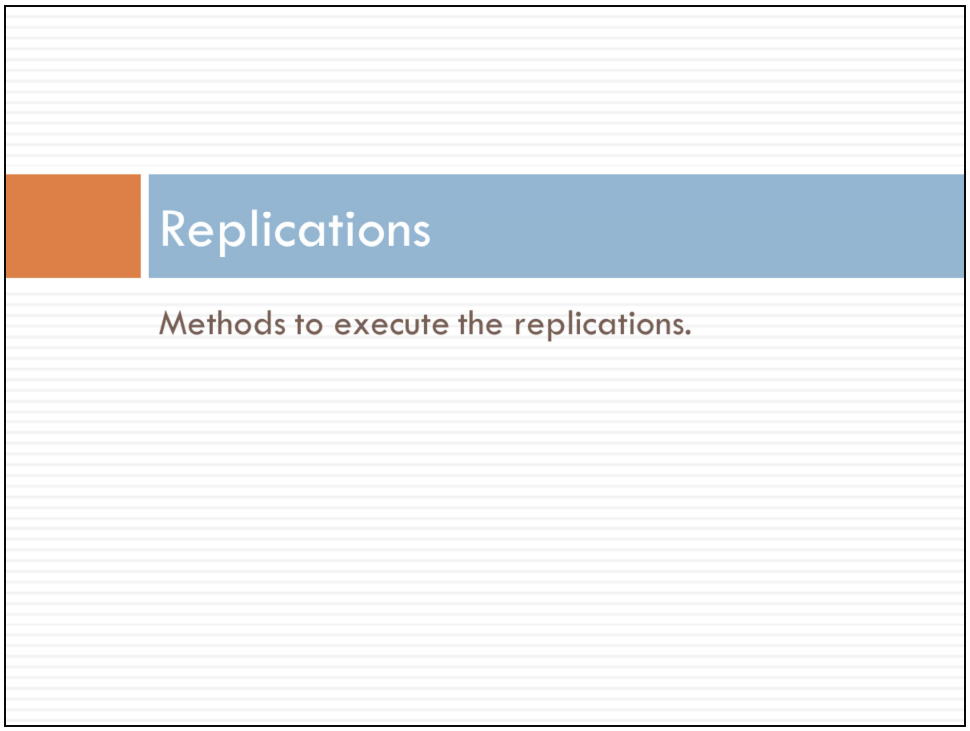
Exercise 3

- Calculate the amount of replications needed for the Exercise 2

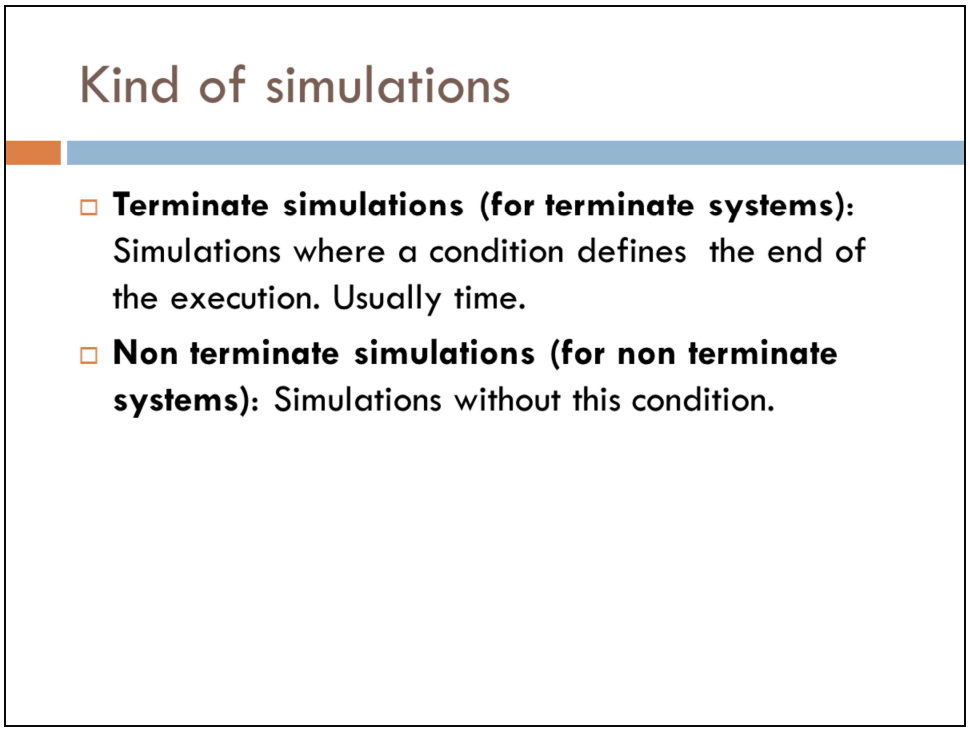
Cleaner	Machines	Workers	x1	x2
-	-	-	20,885	20,261
-	-	+	33,836	36,368
-	+	-	9,9099	142
-	+	+	17,766	131,13
+	-	-	42,759	0,0402
+	-	+	5,7025	2,327
+	+	-	10,481	8,7404
+	+	+	5,9775	5,1167

	1 − α							
r	0.75	0.80	0.85	0.90	0.95	0.975	0.99	0.995
1	1.000	1.376	1.963	3.078	6.314	12.706	31.821	63.657
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106
12	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055
13	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947
16	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921
17	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898
18	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861
20	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845
21	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831
22	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819
23	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807
24	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797
25	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787
26	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779
27	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771
28	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763
29	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756
30	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750
40	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704
60	0.679	0.848	1.046	1.296	1.671	2.000	2.390	2.660
120	0.677	0.845	1.041	1.289	1.658	1.980	2.358	2.617
∞	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576

Remember the Excel spreadsheet?, can you calculate the number of needed replications for this example?. In this case we start with 2 initial replications.



On previous slides we discuss regarding the number of replications we need to assure that the answer is useful for our needs, this means that the confidence interval we obtain is narrow enough to make good decisions. In this section we will discuss how we are going to execute the different replications, a decision that must be taken at the beginning of the experimental design in order to assure that the results makes sense.



First, it will be needed to understand that we will have two types of simulations, that are referred to two types of elements that we want to analyze, terminate and non terminate systems.

The first, the terminate system, is a system where a specific state is going to happen continuously. This state can represent the end of the system or a specific situation where all the system variables are, again, equal to a previous value. In that case we will use terminate simulations.

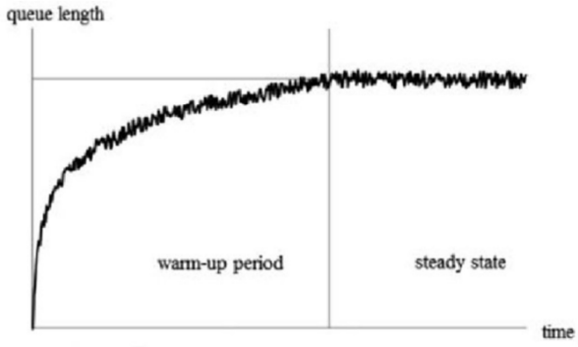
An example of this kind of systems can be a shop, that opens at specific time and closes at other specific time.

The second case in non-finite systems, that implies use non terminate simulations, where there is not a cyclic state, or maybe we don't know it, or maybe is out of the bounds of our analysis.

An example of this kind of simulations is a police station, a hospital, or a factory (a critical one that must always be working). In this cases the system

Is always working, and is not a situation that will be, at least apparently equal to a past situation in all the system variables we analyze.

Transient period



G. Chen and Z. Z. Yang, Methods for estimating vehicle queues at a marine terminal: A computational comparison. Int. J. Appl. Math. Comput. Sci. **24**, 611 (2014).

The transient, loading or, as is shown in the figure, warn-up period, is the period where the answer does not have its real value, this typically happens in a non terminate simulations, because this period is not true. Think in a factory, always you will be able to calculate the mean boxes that you have to be processed by a specific machine. The value 0, maybe is irreal, and exist in the dataset due to the loading process done in the simulation, because starts with no entities.

We will review, depending on the scenario, terminate and nonterminating simulations, how to delete these observations of the transient period.

Independent repetitions

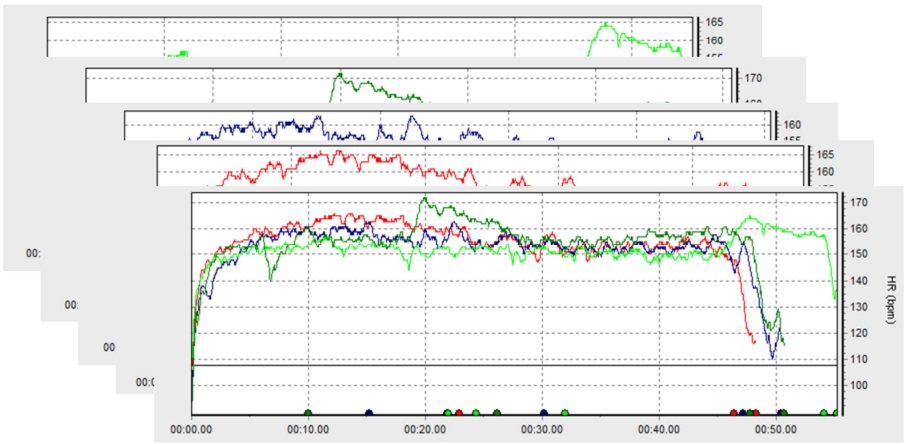
- From the same initial state of the model, that means, with the same parameterizations and behavior, only random numbers to be used un the GAV are changed.
- This different RNG allows test again and again the new system with the different possible values of the variables that are not controlled (random variables).

The first approach to run the different replications is Independent Repetitions. Is the simplest approach, that implies that for each replication we will start from an empty model, and from this empty model, and using a different random number stream, we will obtain a new value for the answer variable.

In this case we define a condition that express that the system is finished, this condition can be represented by asset of conditions in the model variables or can be represented by time (at 8:00 pm the shop closes).

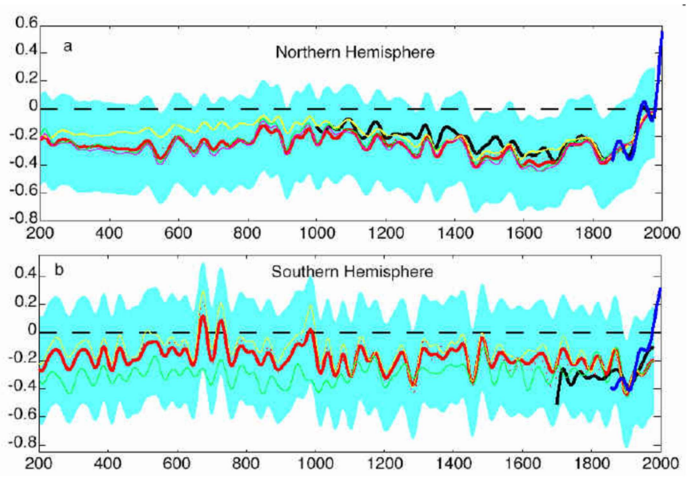
This kind of approach is nice for finite simulations, where we will have a clear condition that defines the end of the replication.

Independent repetitions



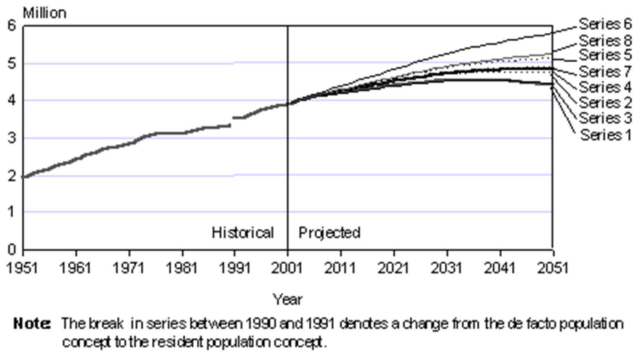
Here you have a set of series that represents some of the different replications we do for a specific answer variable. This model represents the time needed for a runner to finish a 10 kilometers race. As you will notice all the executions are similar, but different due to the randomness.

Independent repetitions



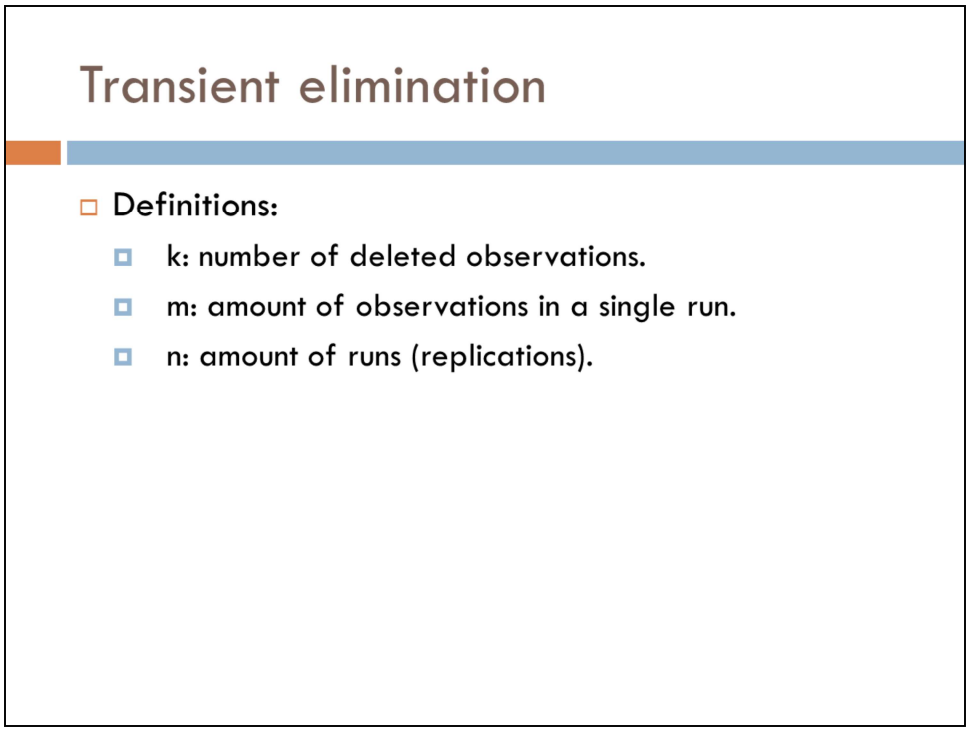
Here we have another example, that tries to analyze the temperatures in the Northern and Southern hemisphere. In that case, the condition that defines when to finish the simulation is a temporal one, artificial from the point of view of the system, but enough from the point of view of the model and the answers we want to obtain.

Independent repetitions



Note: The break in series between 1990 and 1991 denotes a change from the de facto population concept to the resident population concept.

This is another interesting example regarding a hypothetical simulation model that represents the populations in Chicago city. Notice that here we want to predict the future outcome of the city growth. The historical data is going to be used in all the replications to see if the model is working correctly, the, the different replications predicts different outcomes for the city showing the variability. In this case, the use of an independent repetitions approach in interesting because the historical interval we use. This period will be used for the verification and the validation of the model.



But what happens if the transient period is not desired?, in this scenario we can consider that we are on a non-terminal simulation and the transient period makes no sense.

We will start defining the number of deleted observations at the beginning, k.

Then the amount of observations of a single replication, m.

And finally, the number of different replications, n.

Transient elimination

- Step 1: Compute the average of the j observation's over all runs.
  - $\bar{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{i,j}$
- Step 2: Compute the overall average.
  - $\bar{\bar{x}} = \frac{1}{n}\sum_{j=1}^{n} \bar{x}_j$
- Step 3: K=1

With this parameters, the algorithm works as in presented on this and the next slides.

First, we calculate the average of the j observation's over all runs.

Then we calculate the overall average.

And finally we set k to one.

Transient elimination

- Step 4: Compute the overall average of the j observation's over all runs without the first k observations.
  - $\overline{\overline{x_k}} = \frac{1}{n-k}\sum_{j=k+1}^{n} \bar{x}_j$
- Step 5: Calculate the relative change
  - $\Delta = \frac{\bar{\bar{x}}_k - \bar{\bar{x}}}{\bar{\bar{x}}}$
- If $|\Delta_k - \Delta_{k-1}|$ > threshold then increment k and go to step 4, else remove k observations and use

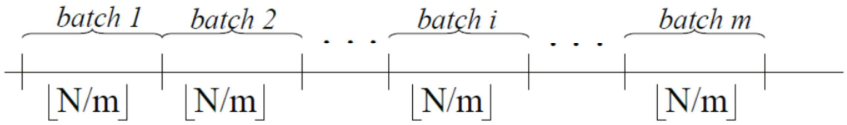
Next, we compute the overall average of the j observation's over all runs, but we do not consider the first k observations.

The wee can calculate the relative change

And finally, we analyze the threshold, if this is bigger that a specific value we define, a percent of the answer we obtain, 5%,as an example, then we will increment k and go to step 4, else we finish, we can remove k observations and use the remaining observations for the analysis.

Batch means

- Execute a long simulation and then divide it in different blocks, or execution bags "batches".
  - We work with the mean values of these observations.
  - The size of the "batches" is $n = \lfloor N/m \rfloor$
- Each one of these observations are considered as independent.
- Is desirable to determine what must be the required long of each one of these execution blocks, to assure the correctness of the experiment.

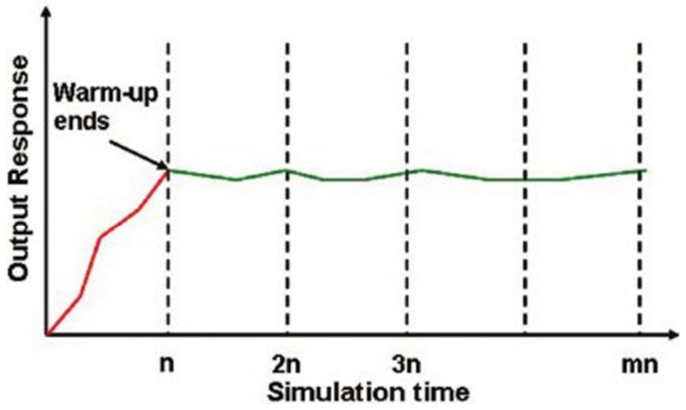


The second method we will review to do the replications is the Batch Means. On this method we will execute a very long simulation, as an example if we are simulation Amazon, we will execute several years, and each one of this years will be a replication. Each one of this years will represent a Batch.

But notice that in this approach the replication we execute is only one, but at the end of each batch (a year in the example) we consider that we have one replication, hence the set of observations that belong to this batch are going to represent a replication.

Batch Means implies to define a method to manage the random number streams that will be more sophisticated that in the "Independent Repetitions" approach, since we must be able to calculate when the random number stream finish, or the Batch, in order change the stream.

Batch means



G. Chen and Z. Z. Yang, A review of techniques for the analysis of simulation output, Int. J. Appl. Math. Comput. Sci. **24**, 611 (2014).

Notice that in Batch means approach the warn-up period is set only one, hence it seems that for non-terminating simulations this method is going to work well. In non-terminating simulation the loading period is not real, hence we must delete if from each one of the different replications we have.

Transient elimination

- Step 1: Set n=2
- Step 2: Compute the average of the "i" batch.
  - $\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{i,j}$
- Step 3: Compute the overall average.
  - $\bar{\bar{x}} = \frac{1}{m}\sum_{i=1}^{m} \bar{x}_i$
- Step 4: Compute the variance of the batch means.
  - $Var(\bar{x}) = \frac{1}{m-1}\sum_{i=1}^{m}(\bar{x}_i - \bar{\bar{x}})^2$

Definitions:
n: amount of observations in a single run.
m: amount of runs (replications).

Like the method proposed to do the transient elimination for Independent Repetitions, the Btch Means owsn a method that is summarized on this and the next slide.

First, we use two definitions, n that is the amount of observations in a single run and m, that is the amount of runs (replications).

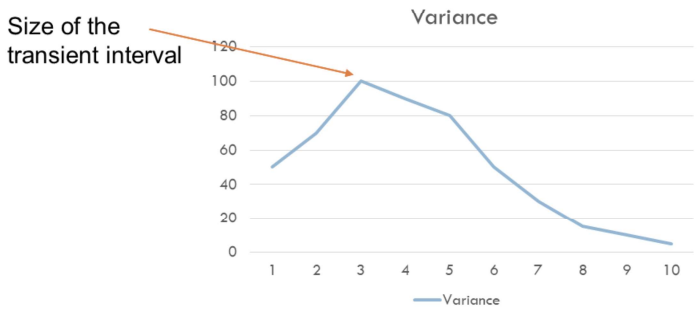
And the we follow the steeps proposed on this slide.

First we set n to two, next we compute the average of the "i" batch.

Next, we compute the overall average. Next we compute the variance of the batch means.

Transient elimination

- Step 5: Increase n by 1 and go to step 2 and plot the variance as a function of n. The point at which the variance starts to decreases is the length of the transient interval.



The we increase n by 1 and go to step 2 and plot the variance as a function of n.

Notice that we review the plot and at the point at which the variance starts to decreases is the length of the transient interval, the values we must delete to correct the analysis.

Regenerative methods

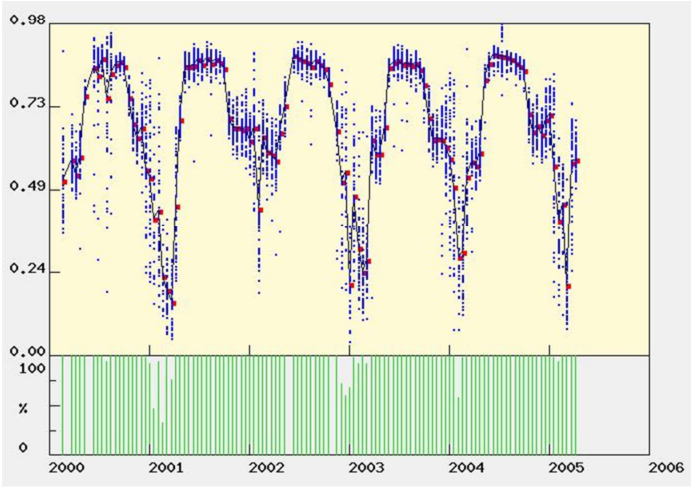
- If the variables observed in the execution of the simulation model, represents, in some way a cyclical restart, that allows suppose the existence of cycles (in the life of the variable). Is likely to consider each one of theses cycles as a replication
- This method is not always applicable. Depends on the existence of cycles in the variables. Also the longitude of this replications must be small; if the longitude of this cycles is big we obtain a small sum of replications.

The last method we are going to present is the regenerative method. Is a derivation of the Batch Means approach, where the definition of the batches are going to be defined depending on the system nature.

On this method we are going to analyze the system and we are going to find those conditions that implies that the system is in a specific stated. Notice that this specific state, name A, is not going to be defined, at least only, by specific values on the system variables. Usually a set of conditions defines this state, that Is the regeneration state of the system.

This method can be applied if this cyclical behavior of the system exists and if we have an strong understanding of the system, that allows us to detect this regeneration points.

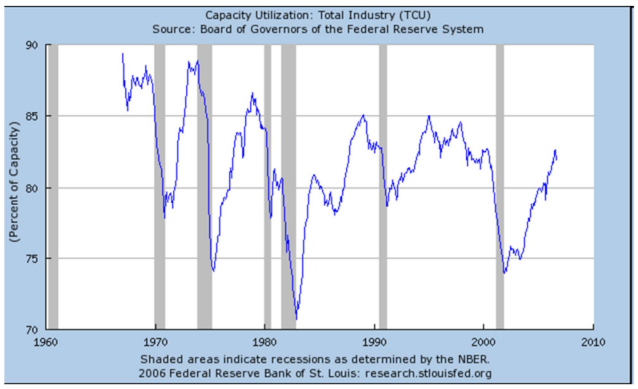
Regenerative methods



Let's go to see this example. As you can see on the picture, the answer variable seems to behave in cyclical pattern. Think as the lowest values of the series, the points where the system regenerates, hence we can consider that when we arrive at this point, a replication is finished.

Notice that in the picture the regeneration points are not showing the same values every time, hence the difficulty to detect this regeneration points.

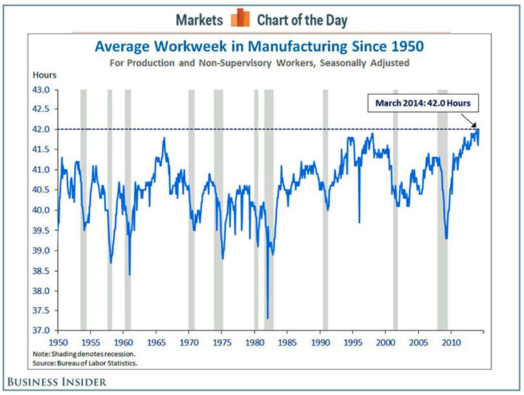
Regenerative methods



This is another example applied to economics where we want to detect the regeneration points. The grey bands detects a crisis, that implies a new economical cycle.

Regarding the chart

- Nobel laureate and Yale University economist Robert Shiller is in the camp of experts who believe the odds of a recession are very low.
  - Read more: http://www.businessinsider.com/shiller-chart-shows-why-recession-is-years-away-2014-4#ixzz30IcrlnI2



Here you can see a new version of the chart. A newest is coming.

Regenerative methods

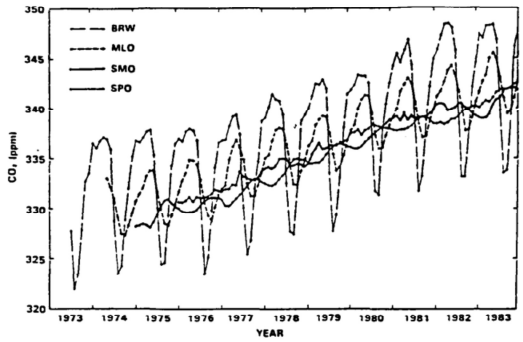


Fig. 1. Selected monthly mean carbon dioxide concentrations from continuous measurements (Barrow, Alaska (BRW); Mauna Loa, Hawaii (MLO); American Samoa (SMO); South Pole (SPO). From: WMO, 1985.

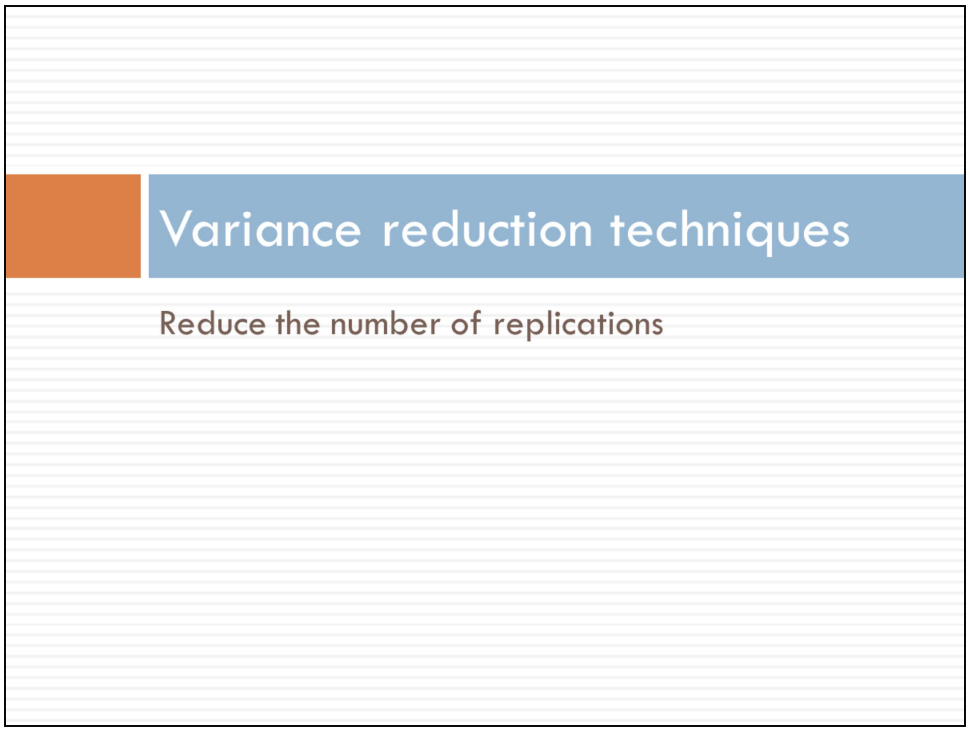
Another interesting example, in that case or the CO2 concentrations. One can see clear the cyclical behavior of the system, but how the levels of CO2 increases every year.

Applicability

	Terminating simulations	Nonterminating simulations
Loading period needed	Independent repetitions	Independent repetitions
Loading period unneeded	Independent repetitions erasing the loading period/ Batch means	Batch means

On this last slide you can see a table that summarizes when to use one method or other, depending of the kind of simulation. The need of use the loading period depends on the system, the answer we want to analyze, and also the kind of experiment we are conducting.

Regenerative methods usually perform better that other alternatives, but as one can see, implies a deeper understanding of the system and a deeper analysis of the model, to be able to detect the regeneration points.



At this point we see that the number of replications that must be done in each scenario is a key element to obtain a confidence interval that is enough for our needs. Now we will review some techniques that can be applied to reduce the number of replications needed in each scenario.

Motivation

- Interest to reduce the variability introduced in the answer variable due to the use of RNG.
- The value that estimates a specific answer variable, that is represented by its confidence interval, must be adjusted (as possible).

$$(\bar{x} - k\, s/\sqrt{n}, \bar{x} + k\, s/\sqrt{n})$$

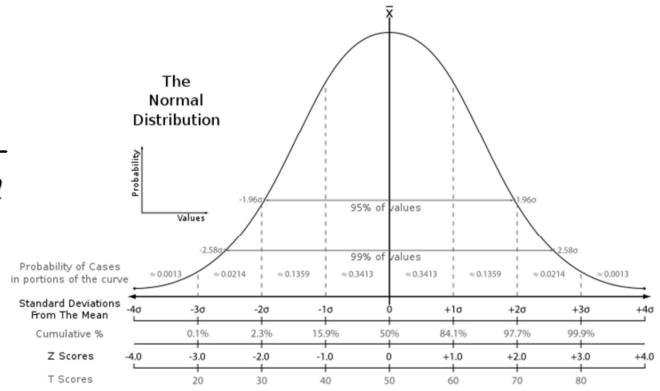
It is clear the only way to reduce the confidence interval is to increase the number of observations, replications. However this approach sometimes can be expensive, since we do not have enough resources to do the desired replications to narrow the interval.

The other alternative we have is to try to reduce the variance we observe in the different observations. To do so, we can try to use the next techniques.

Motivation

- Obviously, increasing n, that is the number of observations, the standard error decreases. Variance reduction techniques try to reduce this variability without the need of increase the number of observations.

$$s/\sqrt{n}$$



In brief, the justification of the use of this techniques are presented on this slide. Notice that if we can obtain values that copse better the variability of the normal distribution, the number of observations will be reduced.

Antithetic variables

- Use of antithetic values o the random numbers stream used.
- In the first execution the random numbers used can be (a, b, c, ..) ∈ [0,1).
- In the second execution we use its antithetic values, that means (1-a, 1-b, 1-c, ..) ∈ [0,1).
- Is needed to establish a synchronization method between both streams

The first technique that we can apply to reduce the number of replications is the antithetic technique. In this technique we will notice that the generation of any Random Variable distribution, rely on a value generated from a Random Number Distribution, that is a Uniform Distribution, with values between 0 and 1.

Notice that, if we subtract to one, any number generated by this distribution, we obtain again a Uniform Distribution, that can be used to generate Random Variables. This second value we obtain is the antithetic variable. Notice that we expect that the first random value generates a set of numbers that “explore” a set of the space variables, while the antithetic variable will generate a set of numbers that “explore” the symmetric space. With this idea we expect to find faster values, answers, that are in the opposite sides of the normal distribution that defines the answer.

The method to use the antithetic variables is quite simple and in presented on this slide.

In the first execution the random numbers used can be the normal random numbers generated by the RNG.

In the second execution we use its antithetic values, that means 1 minus RNG. Remember that to use any stream we must test it.

In order to use this in your simulator, it will be needed to establish a

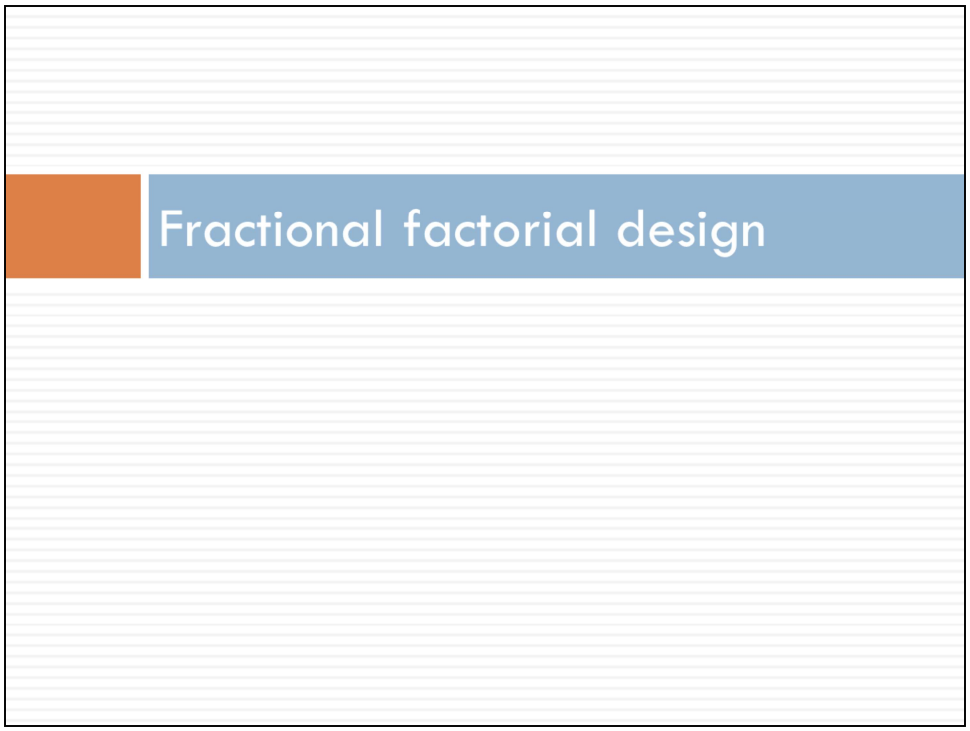
synchronization method between both streams, to assure that we are going to change to the antithetic RNG when needed.

Control variables

- Simulation allows the observation of the system evolution during the execution of the experiment.
- This allows, in certain grade, to compare the values of the answer variables with the observed values.
- We can add modification to the model to reduce the difference.

A control variable is an experimental element which is constant, hence we suppose that if all the execution is going correctly, it will remain with the same value during all the scenario execution. This means that this control variable often is really a constant, think as an example if you know that certain parameter of the experiment must remain constant. Often however, this control variables depends on other parameters. A modification on this value implies that something is wrong with the assumption, often because the simplification assumptions are so strong.

From the point of view of the experimentation, obviously the control variables are not interesting, but its use can reduce the number of replications needed, since when we detect that it changes, is because something is wrong, and we can discard, or modify the course of the scenario execution.



At this point we can reduce the number of scenarios to consider constraining the number of levels to two, also we know some techniques to try to reduce the number of replications needed to obtain the confidence interval. However, the number of needed scenarios can be huge if the number of factors to analyze are huge.

In this part of the course we will see some techniques to reduce the number of scenarios to consider, trying to keep the power of our analysis as intact as possible.

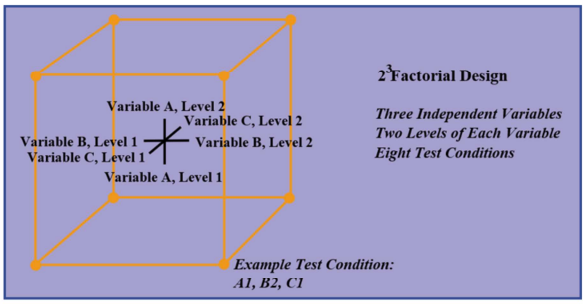
Fractional factorial design

- A factorial experiment in which only an **adequately chosen fraction** of the treatment combinations (scenarios) required for the complete factorial experiment **is selected to be run.**

With this idea, we present the “Fractional Factorial Designs” where we will define a method to select a part of the scenarios to be executed in a full factorial design, selecting adequately this chosen fraction of the scenarios in order to obtain as maximum information as possible.

A 2^{3-1} design (half of a 2^3)

- Consider the two-level, full factorial design for three factors, namely the 2^3 design. This implies eight runs (not counting replications or center points).



By Nicoguaro - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=12758102

Look at the cube presented on this slide. This resents the space that a full factorial 2 to 3 design is going to explore. Each one of the vertices is an answer we obtain, due to a specific combination of the tree factors that we have in this scenario.

Notice that is we want to understand the shape of the figure we are going to obtain, only 4 points, accurately selected, will be needed. This idea can be formalized next, defining how to select properly these points, these scenarios.

2^3 Two-level, Full Factorial Design

	X1	X2	X3	Y
1	-1	-1	-1	$y_1 = 33$
2	+1	-1	-1	$y_2 = 63$
3	-1	+1	-1	$y_3 = 41$
4	+1	+1	-1	$Y_4 = 57$
5	-1	-1	+1	$y_5 = 57$
6	+1	-1	+1	$y_6 = 51$
7	-1	+1	+1	$y_7 = 59$
8	+1	+1	+1	$y_8 = 53$

On this slide you can see a full factorial design for a 2 to 3 design. From this design we know that we can calculate the main effects and interactions using Yates algorithm or the handy formulas.

Computing the effects

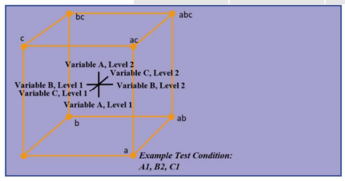
- Effect of X1= (1/4)(y2 + y4 + y6 + y8) - (1/4)(y1 + y3 + y5 + y7)
- X1 = (1/4)(63+57+51+53) - (1/4)(33+41+57+59) = 8.5
- Suppose, however, that we only have enough resources to do four runs. It is still possible to estimate the main effect for X1? Or any other main effect?
  - The answer is yes, and there are even different choices of the four runs that will accomplish this.

Here we have the expressions needed to calculate the main effect for X1, the first factor. We want to know is it is possible to obtain this information but without all the scenarios execution. Think that we have only resources to do four executions. We may answer if it is still possible to estimate the main effect for X1? Or any other main effect?

Hopefully, the answer is yes, and there are even different choices of the four runs that will accomplish this. The selection of this choices is what drive the fractional factorial design approach.

Only 4 runs

		C1	C2	C3	Y
Y1	1	-1	-1	-1	$y_1 = 33$
Y2	2	+1	-1	-1	$y_2 = 63$
Y3	3	-1	+1	-1	$y_3 = 41$
Y4	4	+1	+1	-1	$Y_4 = 57$
Y5	5	-1	-1	+1	$y_5 = 57$
Y6	6	+1	-1	+1	$y_6 = 51$
Y7	7	-1	+1	+1	$y_7 = 59$
Y8	8	+1	+1	+1	$y_8 = 53$



Notice that we can select this fraction of the scenarios we have, the one selected on brown color can be a selection, and from this selection we can try to calculate the main effects for C1, C2 and C3. notice that this selection is like select a subset of the points of the cube. Are this selection enough to be able to calculate this main effects?

Main effects

- C1 main effect:
  - c1 = (1/2) (y4 + y6) - (1/2) (y1 + y7)
  - c1 = (1/2) (57+51) - (1/2) (33+59) = 8
- C2 main effect
  - c2 = (1/2) (y4 + y7) - (1/2) (y1 + y6)
  - c2 = (1/2) (57+59) - (1/2) (33+51) = 16
- C3 main effect
  - c3 = (1/2) (y6 + y7) - (1/2) (y1 + y4)
  - c3 = (1/2) (51+59) - (1/2) (33+57) = 10

In this slide we can notice that the selection we propose on the previous slide works, since we are able to calculate the main effects of the different factors we have.

But we can calculate the interactions?. With this we are not going to be able to calculate this. Hence the question we want to answer is, how we will select the fraction of the scenarios to be able to obtain as maximum information as possible?

Selecting the experiments to execute

- Note that, mathematically, $2^{3-1} = 2^2$

	X1	X2
1	-	-
2	+	-
3	-	+
4	+	+

To select the scenarios to be executed we will follow an approach that largely simplifies the process. First notice that this table represents a two to two factorial design, and that this is equivalent, numerically, to a two to three minus one.

This represents that we have only four different scenarios and that those scenarios represents all the combinations of the two factors that we are representing on the table.

Adding the column of the interactions

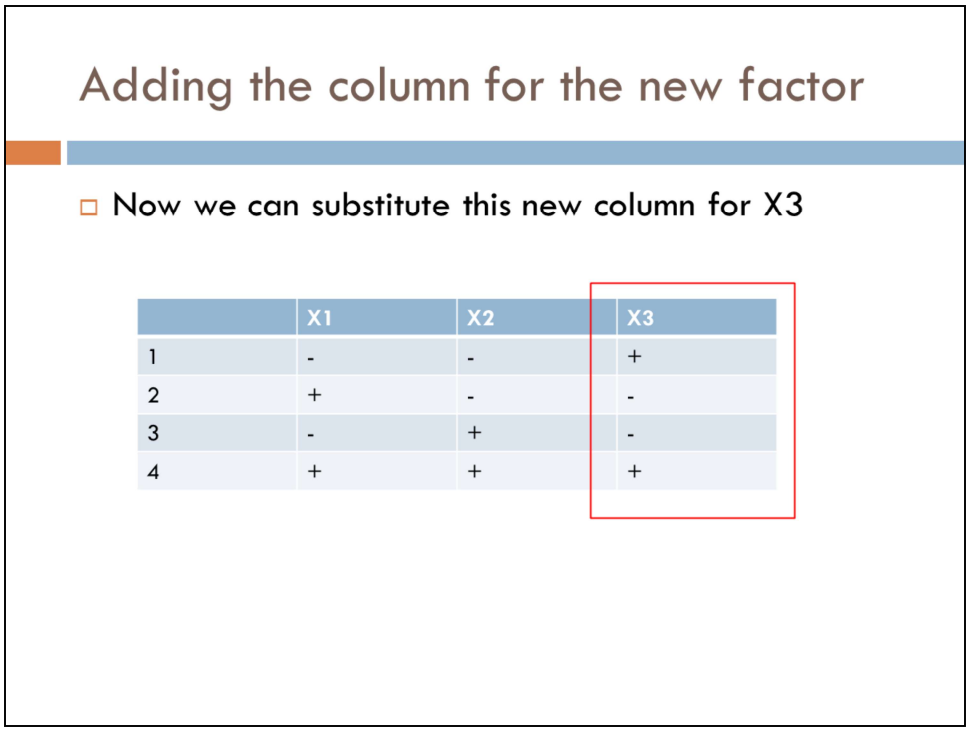
- We add a new column that represents the interactions between X1 and X2

	X1	X2	X1*X2
1	-	-	+
2	+	-	-
3	-	+	-
4	+	+	+

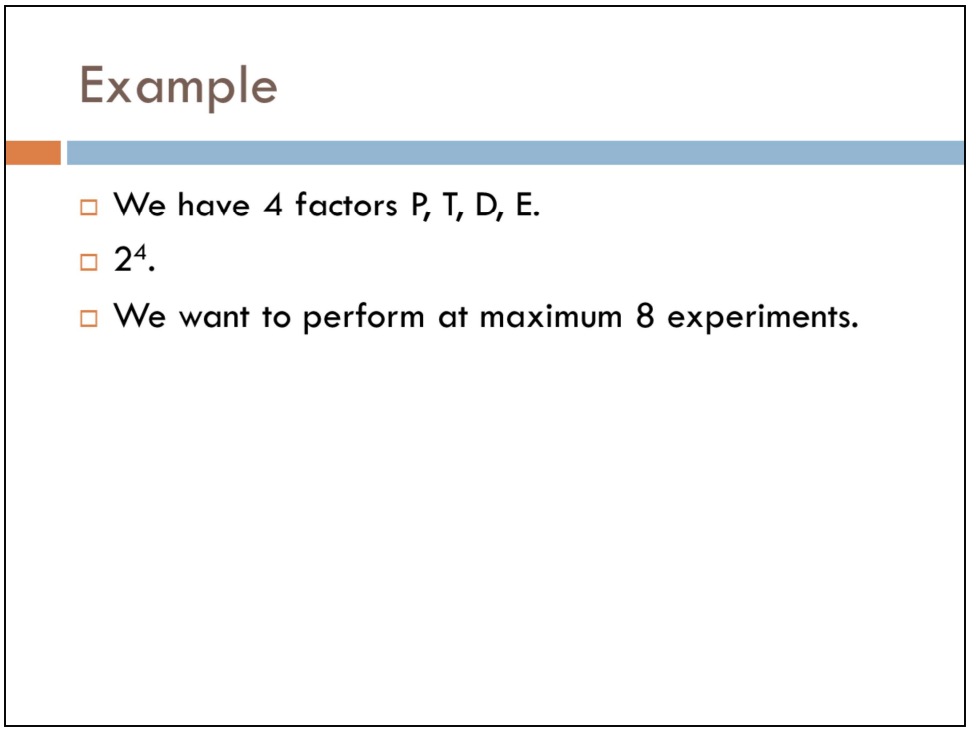
Since we want to analyze three factors, not only two, we can add a new column that represents this new factor. The value, the level, we will use in each scenario, will be calculated using the levels of the other two factors, as the multiplication of the plus or minus representation. This implies that if we have two plus signs, or two minus signs, the level we will use in the new column will be plus, minus otherwise.

On the table is presented this new column with the correct value from the combination of the existing factors levels.

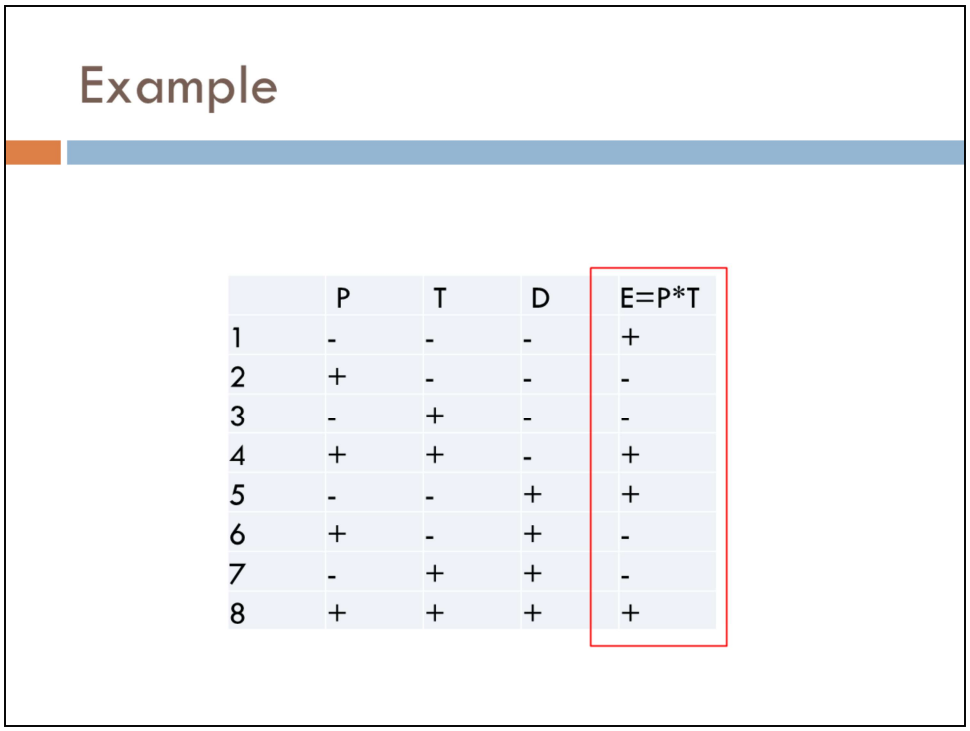
Notice that the semantics are important to fix what means minus or plus, but is not relevant to the numerical solution, since this solution depends on the definition of a geometrical space. This geometrical space will be fixed at the beginning with the definition of the semantics on the levels and constrained with the construction of the fractional factorial design.



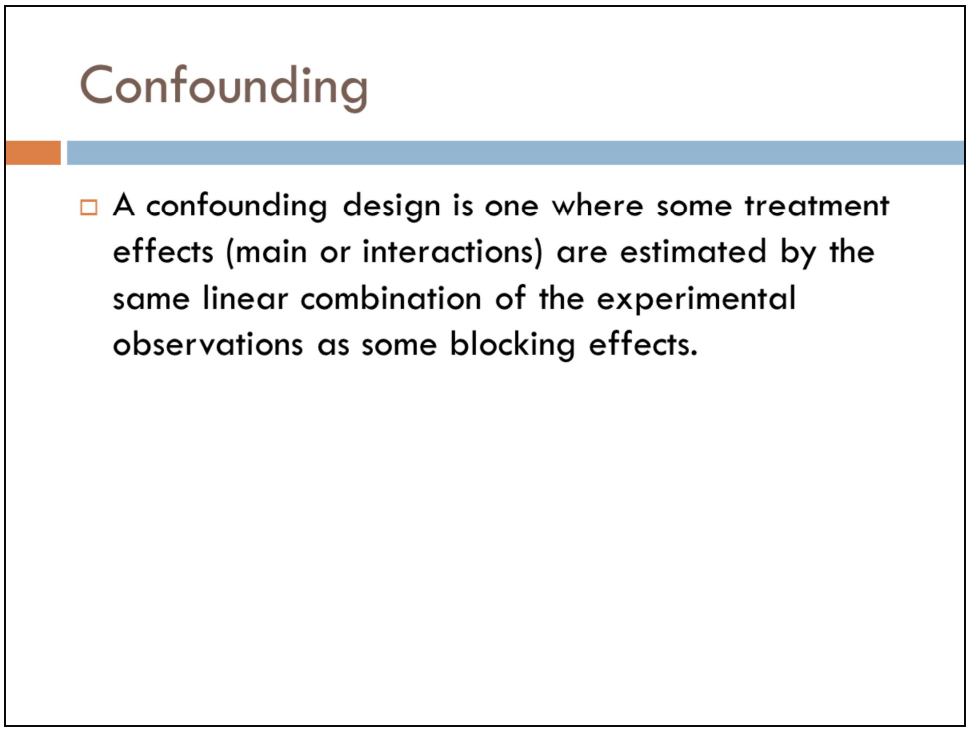
Once he have the table, we can change the caption of the column for the name of the factor, and we have the experiment ready to be executed. Notice that we add a new column, but we are not adding more rows, scenarios.



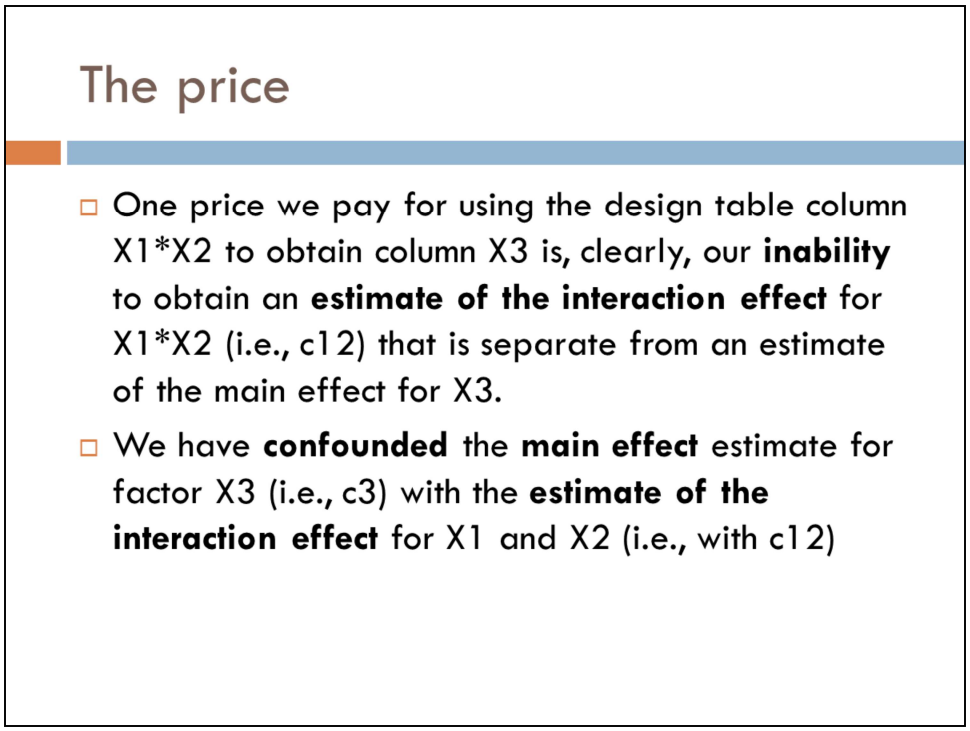
Let's go to review now an example. Imagine that we have four factors, named P, T D and E. If we want to analyze the effects and interactions on theses factors, the first approximation we will follow will be a factorial design. Then, the number of scenarios to consider will be two to four. This implies that we have 16 different scenarios to execute. But, consider that we have resources to execute only 8 scenarios, we don't care regarding the needed replications at this point.



Since we need to reduce the experiments to 8, we will define a full factorial design for three factors, generating two to three, hence 8, different scenarios. The last column will be defined by a combination of factors P and T, as we can see on the table.



The column that we are adding to the table is following a confounding pattern, in that case E equals to P multiplied by T. A confounding design is defined as one where some treatment effects, that will be main effects or interactions will be estimated by the same linear combination of the experimental observations as some blocking effects. That is that the combination we use on the confounding patter will be used to explain the main effects and combinations of the factors we have.



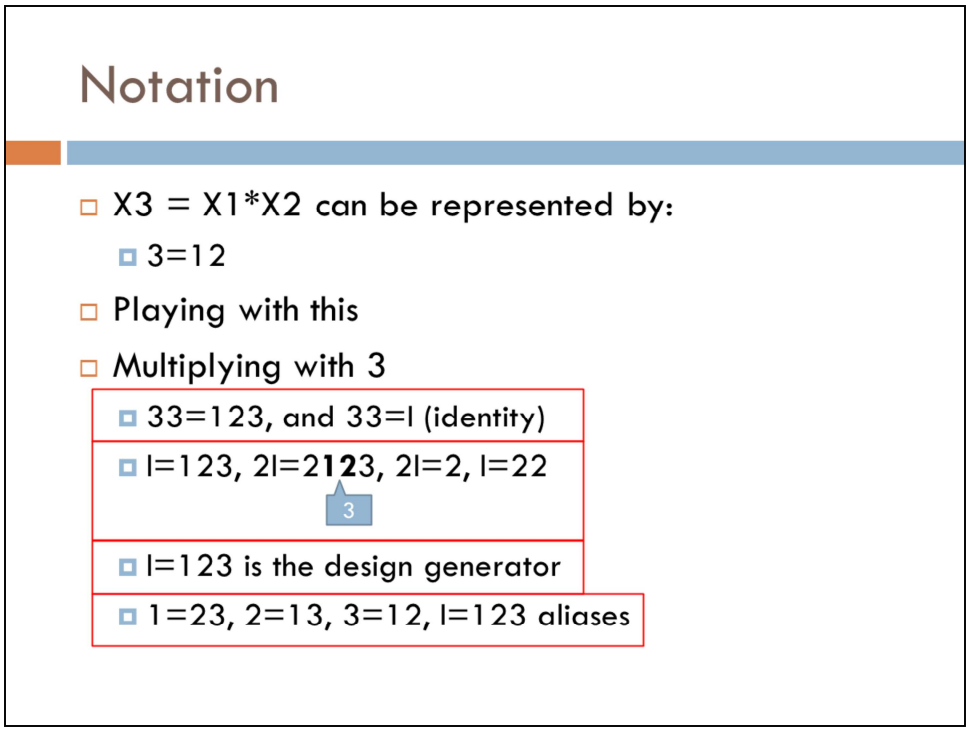
As one can imagine, the use of a confounding patters, hence, to reduce the number of scenarios to be executed, is not free, and implies that there are some information that we are going to loose.

As an example, the price we pay for using the design table column X1 multiplied by X2 to obtain the column X3 is, clearly, our inability to obtain an estimate of the interaction effect for X1 and X2, because it is distinct from the estimate of the main effect for the factor X3.

Notice that we have confounded the main effect estimate for factor X3, with the estimate of the interaction effect for X1 and X2, hence we loose the information we drawn on the patters, the interaction between X1 and X2.

This price can be seen as high, however, depending on the pattern you are going to use will imply loose interactions of several factors, that maybe are not of the interest of the analysis. Notice that the variability of the interactions of several factors, maybe is high enough to make no sense to use them for any conclusion.

This is the price we pay for the reduction in the number of scenarios.



The notation we will use to define this fractional factorial designs generators are presented on this slide. See that we simply represent the factor by a number, one, two, and so on.

Notice also that we can see that a factor, multiplied by itself is returning the identity, since we will obtain always a matrix of one, plus sign.

With this idea we can operate the values and do some simplifications if needed, on the definition of the designs.

Also we can define a patters ad the design generator, that is the one that can be used to generate any other pattern.

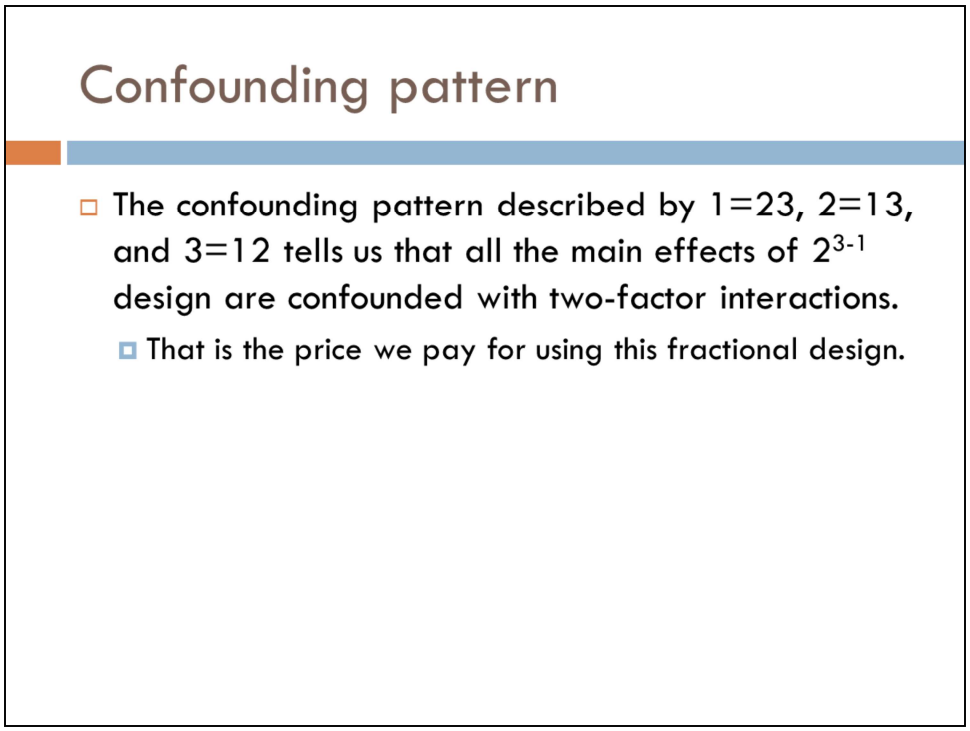
Finally, just to mention that exists different aliases that can be used for the experiment. The decision of what will be used depends on the price, our inability to detect some interactions, we want to pay.

Principal fraction

- We can replace any design generator by its negative counterpart and have an equivalent, but different fractional design.
- The fraction generated by positive design generators is sometimes called the principal fraction.

It is worth to mention also that we can replace any design generator by its negative counterpart, doing so we will obtain an equivalent, but a different fractional design.

The fraction generated by positive design generators is sometimes called the principal fraction.



Just to remark again that the confounding pattern described by 1=2 and 3, 2=1 and 3, and 3=1 and 2, tells us that all the main effects of a 2 to 3-1 design are confounded with two-factor interactions.

As we already know, is the price we pay for using this fractional design. Remember that the selection of the specific interaction you lose, depends on the final selection of the factors you use to define the confounding pattern.

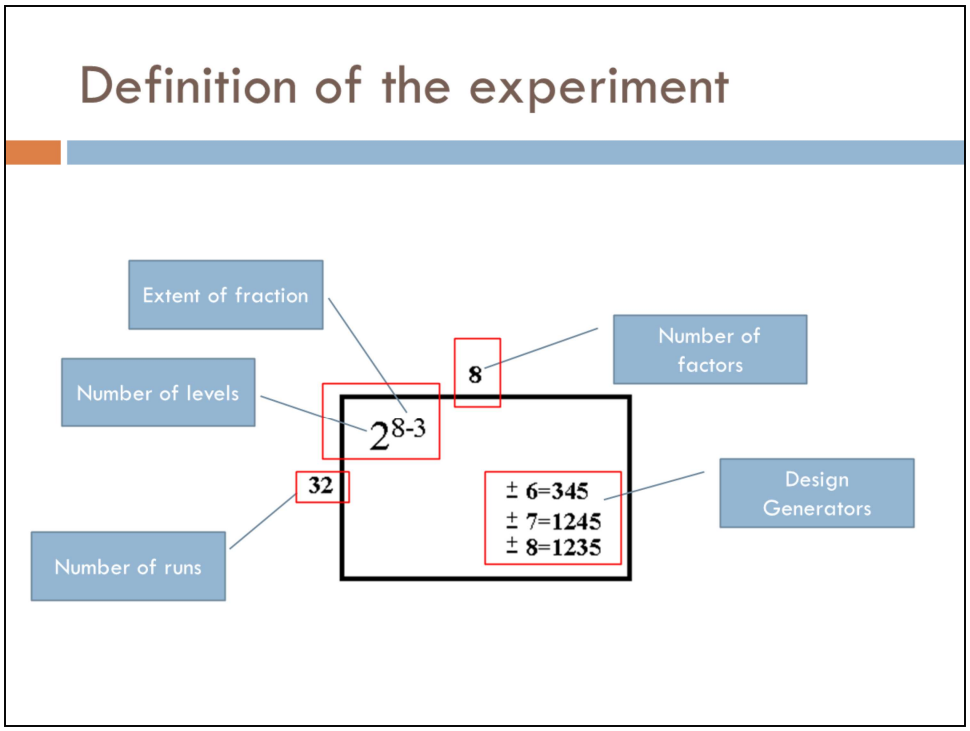
Confounding pattern

- In the typical quarter-fraction of a 2^6 design, i.e., in a 2^{6-2} design, main effects are confounded with three-factor interactions (e.g., 5=123) and so on.
- In the case of 5=123, we can also readily see that 15=23 (etc.), which alerts us to the fact that certain two-factor interactions of a 2^{6-2} are confounded with other two-factor interactions.

Notice that in higher fractional factorial designs, with more factors, the price we are paying implies loosing interactions of several factors. As an example in a 2 to 6 factorial design, if we want to constrain the experiment to 2 to 4 different scenarios, a pattern can be 5 to 1 and 2 and 3, hence we are going to loose information for three factor interactions.

Notice also that 5 to 1 to 2 and to 3 can be transformed dot 1 to 5 = two to 3, with shows clearly that some two-level interactions are confounded with other two factor interactions.

However, depending on the patter we select for both factors to be confounded, notice that the reduction of factors is two, we will not lose this information on the final analysis.



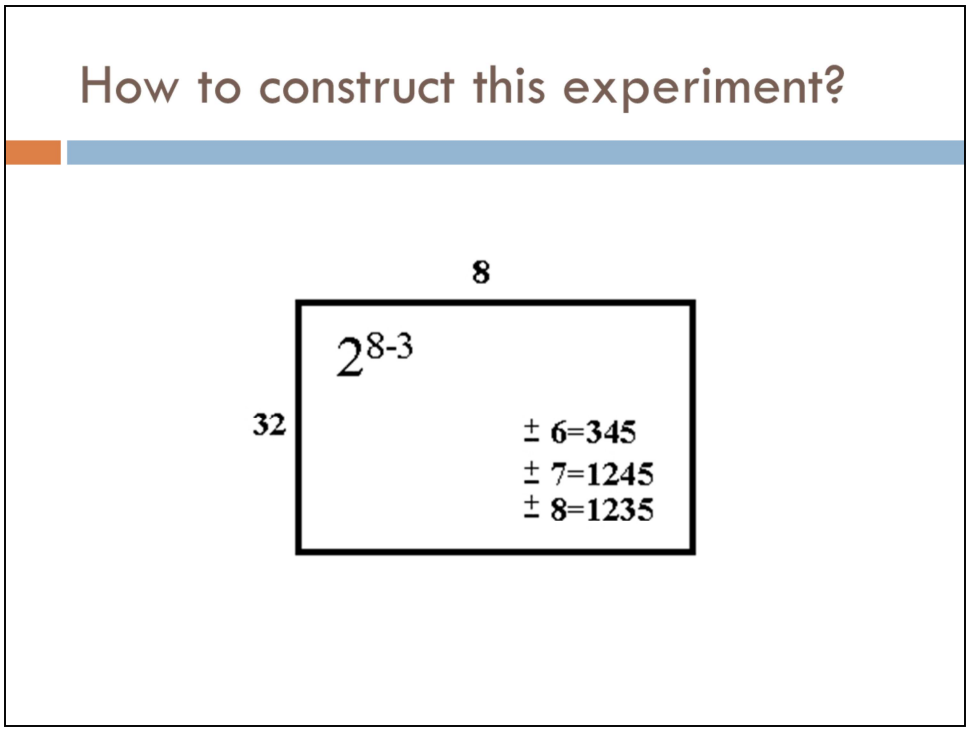
The definition of a fractional factorial design can be represented on this slide. Often this experimental schemas are presented like this, with an square that encompasses all the needed information in order that one, that will be the responsible of the execution of the experiment, clearly understands what to do.

The first element to consider is the definition of the number of factors we have, in this case 8.

Then we define the reduction we do, in that case, we will reduce by three the number of factors.

Because we reduce this by three, we must define the design generators, three.

Finally, we can calculate the number of scenarios to execute. Notice that here the needed replications is not included.



Then with this, we must be able to build an experiment and to understand what is the information we are not going to be able to calculate.

Construct a Fractional Factorial Design From the Specification

- **Write down a full factorial design** in standard order for *k-p* factors (8-3 = 5 factors for the example above). In the specification above we start with a 2^5 full factorial design. Such a design has $2^5 = 32$ rows.
- **Add a sixth column** to the design table for factor 6, using 6 = 345 (or 6 = -345) to manufacture it (i.e., create the new column by multiplying the indicated old columns together).
- **Do likewise** for factor 7 and for factor 8, using the appropriate design generators.
- The resultant design matrix gives the **32 trial runs** for an 8-factor fractional factorial design.

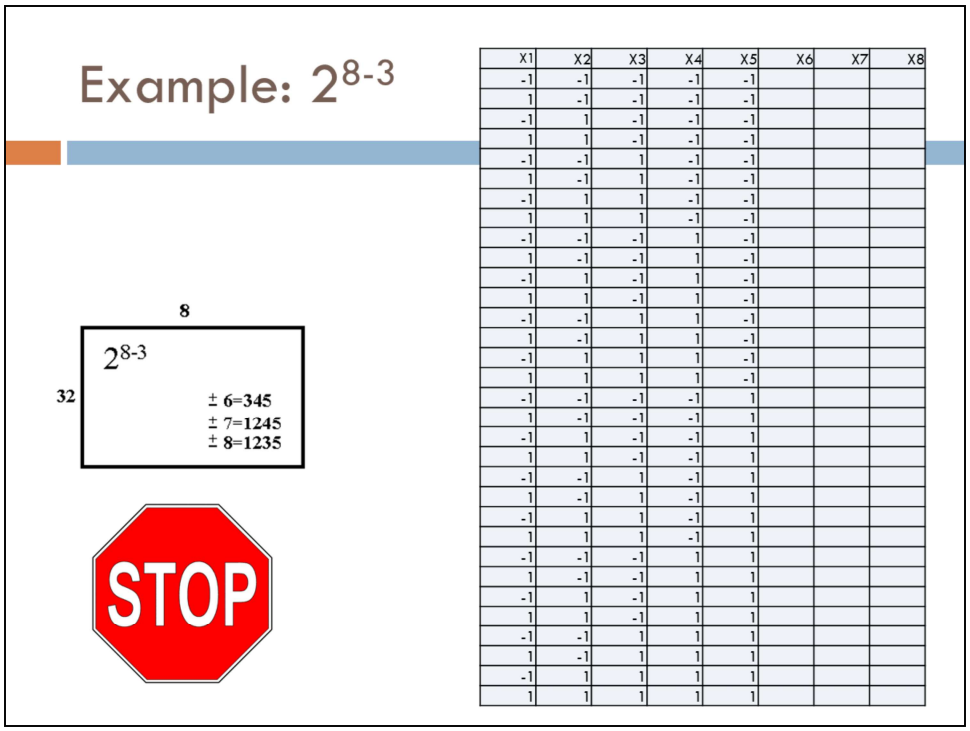
Here you have the detailed description of how to write a fractional factorial design from the pattern.

First, we will write down a full factorial design in standard order for k-p factors, in this case, 8-3 = 5 factors, for the example of the previous slide. Then new have 2 to 5 full factorial design, hence we will start with 32 rows, scenarios.

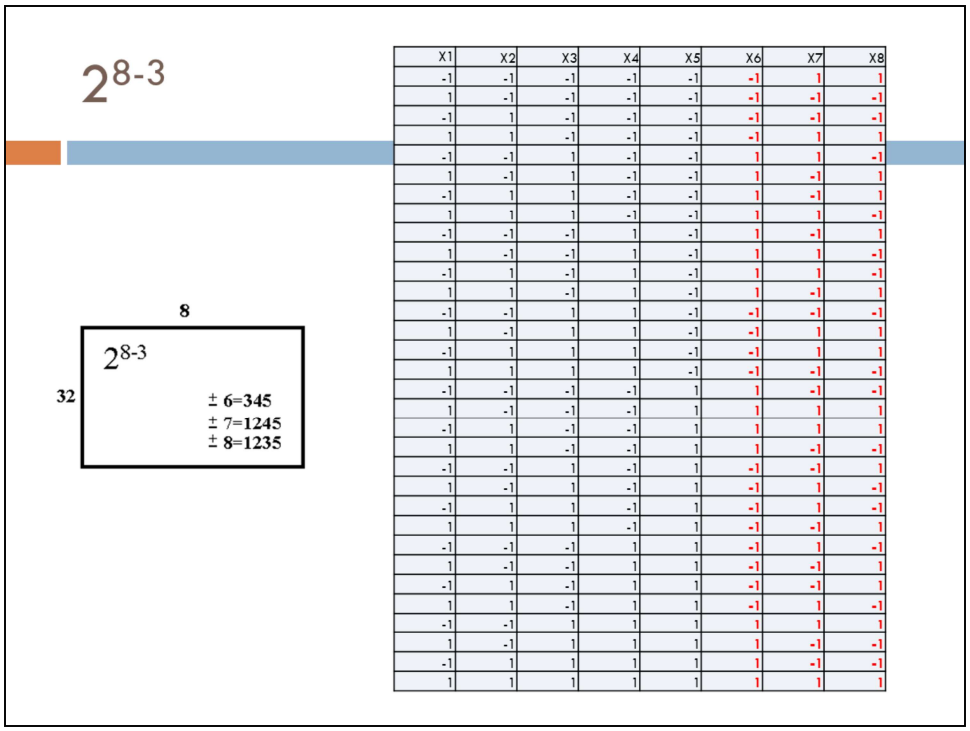
Second, we will add a sixth column to the design table for factor 6, using 6 = 3 and 4 and 5 (or 6 = minus 3 and 4 and 5, that is equivalent) to generate it. Remember that we will create this new column by multiplying the indicated columns together.

Third, and because we only do a confounding for three factors, we do the same for factor 7 and for factor 8, but in that case using the appropriate design generators.

As a result, we obtain a design matrix gives the 32 scenarios for an 8-factor fractional factorial design.



On this slide you have the first steep needed to build the fractional factorial design. Do you know how to fill the table?. Please stop the video, the answer on the next slide.



X1	X2	X3	X4	X5	X6	X7	X8
-1	-1	-1	-1	-1	-1	1	1
1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	-1
1	1	-1	-1	-1	-1	1	1
-1	-1	1	-1	-1	1	1	-1
1	-1	1	-1	-1	1	-1	1
-1	1	1	-1	-1	1	-1	1
1	1	1	-1	-1	1	1	-1
-1	-1	-1	1	-1	1	-1	1
1	-1	-1	1	-1	1	1	-1
-1	1	-1	1	-1	1	1	-1
1	1	-1	1	-1	1	-1	1
-1	-1	1	1	-1	-1	-1	-1
1	-1	1	1	-1	-1	1	1
-1	1	1	1	-1	-1	1	1
1	1	1	1	-1	-1	-1	-1
-1	-1	-1	-1	1	1	-1	-1
1	-1	-1	-1	1	1	1	1
-1	1	-1	-1	1	1	1	1
1	1	-1	-1	1	1	-1	-1
-1	-1	1	-1	1	-1	1	-1
1	-1	1	-1	1	-1	1	-1
-1	1	1	-1	1	-1	1	-1
1	1	1	-1	1	-1	-1	1
-1	-1	-1	1	1	-1	1	-1
1	-1	-1	1	1	-1	-1	1
-1	1	-1	1	1	-1	-1	1
1	1	-1	1	1	-1	1	-1
-1	-1	1	1	1	1	1	1
1	-1	1	1	1	1	-1	-1
-1	1	1	1	1	1	-1	-1
1	1	1	1	1	1	1	1

Here you have the answer.

Words

- There are seven "*words*", or strings of numbers, in the defining relation for the 2^{8-3} design, starting with the original three generators and adding all the new "words" that can be formed by multiplying together any two or three of these original three words. These seven turn out to be I = 3456 = 12457 = 12358 = 12367 = 12468 = 3478 = 5678.
- In general, there will be $(2^p - 1)$ words in the defining relation for a 2^{k-p} fractional factorial.

Just to mention that we can see what is the n umber of different combinations that one can obtain, the words or string of numbers in a specific design, as an example for 2 to 8 minus 3.

In this specific case the combinations we obtain, remember that here we are talking of the generators, are seven.

In general, there will be 2 to p minus 1 words in the defining relation for a 2 to k minus p fractional factorial.

Resolution

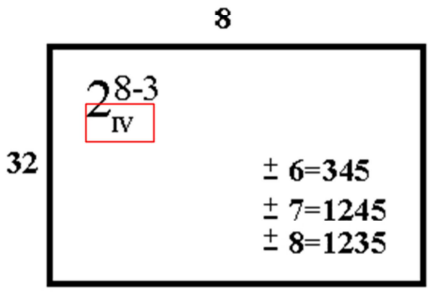
- The length of the shortest word in the defining relation is called the resolution of the design.
- Resolution describes the degree to which estimated main effects are confounded (or aliased) with estimated 2-level interactions, 3-level interactions, etc.
- Resolution is added as a Roman numeral to the experiment definition.

There is an important property of a fractional design named resolution. This is the ability to calculate main effects and interactions from the proposed design. Formally, the resolution of the design is the minimum word length in the definition of the fractional factorial design. Resolution describes the degree to which estimated main effects are confounded, or aliased, with estimated 2-level interactions, 3-level interactions, or below. To understand the resolution in a fractional factorial design, it is added as a Roman numeral to the experiment definition.

	Ability	Example
I	Not useful: an experiment of exactly one run only tests one level of a factor and hence can't even distinguish between the high and low levels of that factor	2^{1-1} with defining relation I = A
II	Not useful: main effects are confounded with other main effects	2^{2-1} with defining relation I = AB
III	Estimate main effects, but these may be confounded with two-factor interactions	2^{3-1} with defining relation I = ABC
IV	Estimate main effects unconfounded by two-factor interactions Estimate two-factor interaction effects, but these may be confounded with other two-factor interactions	2^{4-1} with defining relation I = ABCD
V	Estimate main effects unconfounded by three-factor (or less) interactions Estimate two-factor interaction effects unconfounded by two-factor interactions Estimate three-factor interaction effects, but these may be confounded with other two-factor interactions	2^{5-1} with defining relation I = ABCDE
VI	Estimate main effects unconfounded by four-factor (or less) interactions Estimate two-factor interaction effects unconfounded by three-factor (or less) interactions Estimate three-factor interaction effects, but these may be confounded with other three-factor interactions	2^{6-1} with defining relation I = ABCDEF

Here you have a nice table, from Wikipedia, that defines the implications of the resolution level we use in our experiment.

Complete definition of the DOE



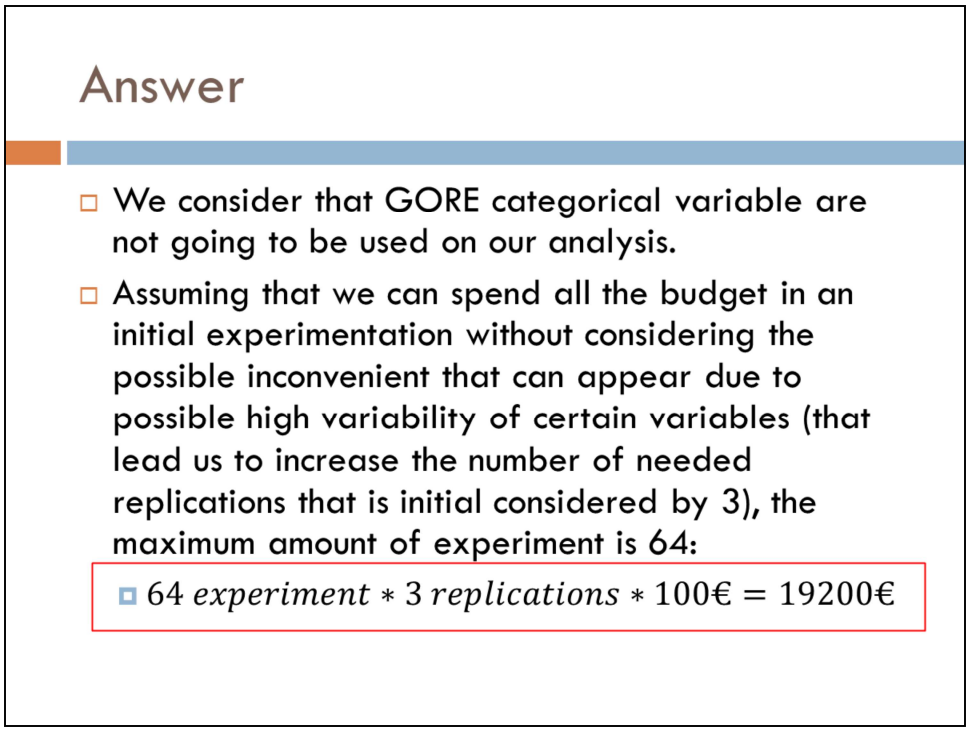
Here you have the same experiment, but now we add the resolution as a roman number.

Example

- We have a limited budget to analyze the different factors to consider on our model. Each individual experiment costs 100€ and we have a total budget of 20.000€ to be destined to experimentation. Define an experiment design, with this constraint, considering that we need at least 3 replications for each experiment.

GORE	BUSH	BUCHANAN	NADER	BROWNE	HAGELIN	HARRIS	MCREYNOLDS	MOOREHEAD	PHILLIPS
ALACHUA	47300	34062	262	3215	658	42	4	658	21
BAKER	2392	5610	73	53	17	3	0	0	3
BAY	18850	38637	248	828	171	18	5	3	37
BRADFORD	3072	5413	65	84	28	2	0	0	3
BREVARD	97318	115185	570	4470	643	39	11	11	76
More than 100 cases...									

Here you have an example. Can you do it by yourself? The answer on the next slides.



We will start understanding that GORE is a categorical variable, hence is not going to be considered in our analysis.

We assume that we can spend all the budget in an initial experimentation. Notice that here we are not considering the possible problems that can appear due to possible high variability of certain variables. This implies the need to increase the number of needed replications that is initial considered by 3., due to the constrain that the maximum amount of experiment is 64.

Answer

- This design don't allow a complete experimentation considering all the factors we have on our model, we have 9 factors that implies a 2^9 experiments meaning 512 experiments with 3 replications each one of them.
- This implies that we need to reduce the amount of variables to be considered using a fractional factorial design.

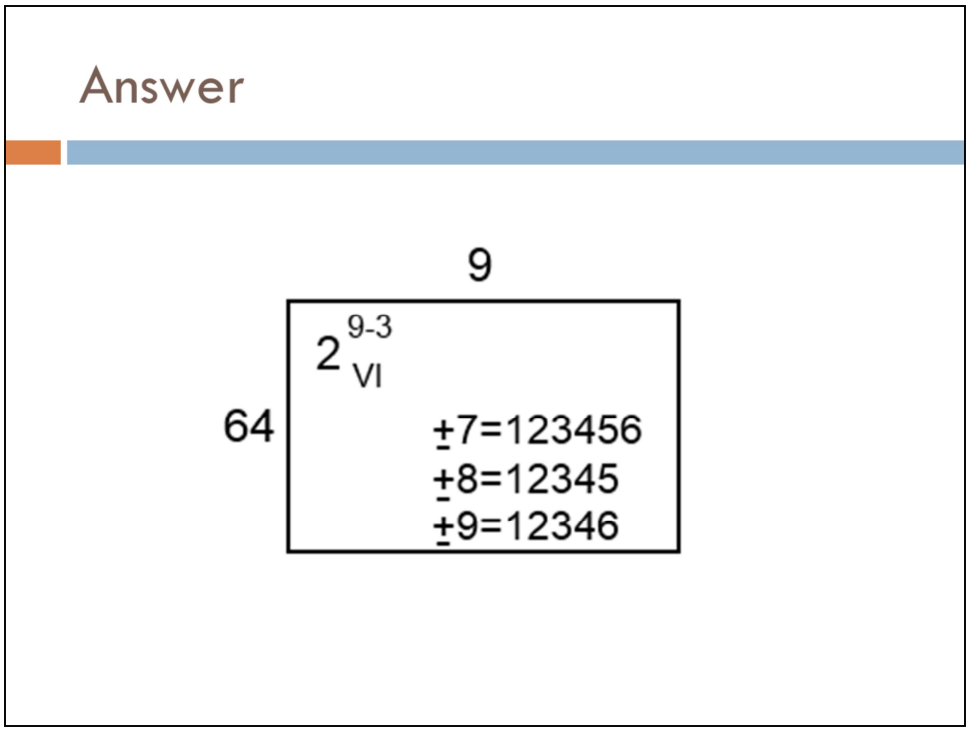
A 2 to 9 design implies a number of scenarios o 512, and also with 3 replications each one of them. This clearly is not affordable in our schema; hence we must use a fractional factorial design.

Answer

- To do this it is needed to define a confounding pattern and select those factors that are going to be confounded.
- The design will be defined as:
  - 2^{9-3}
- Hence 3 confounding patterns must be defined.
- We select in our case the last three variables to be confounded using the first 6 variables.

To do so, we will define a confounding patters, and the design will be designed following the 2 to 9 minus 3. Hence, we have three confounding patters to be defined.

Since there is no preferences, we can use the first 6 variables to be used as a basis to confound the last three.



With all this ideas, the pattern we will follow is this. Do you identify clearly all the elements?.