Data Farming

En omvärldsanalys

JOHAN SCHUBERT, FRIDA HINSHAW

FOI-D--0437--SE December 2011 **Informationssystem**

Johan Schubert, Frida Hinshaw

Data Farming

En omvärldsanalys

Titel	Data Farming – En omvärldsanalys
Title	Data Farming – A survey
Rapportnr/Report no	FOI-D--0437--SE
Rapporttyp/ Report Type	
Sidor/Pages	48 p
Månad/Month	December
Utgivningsår/Year	2011
ISSN	
Kund/Customer	Totalförsvarets forskningsinstitut
Projektnr/Project no	IS916
Godkänd av/Approved by	Jonas Palm

FOI, Totalförsvarets Forskningsinstitut
Avdelningen för Informationssystem

164 90 Stockholm

FOI, Swedish Defence Research Agency
Information Systems

SE-164 90 Stockholm

http://www.foi.se/fusion/

Sammanfattning

Hur kan vi stödja beslutsfattare när det enda som är riktigt säkert är att ingenting är helt säkert? För varje ny uppgift vi ställs inför har vi förmodligen kunskap om vilka faktorer som påverkar, men däremot har vi oftast väldigt liten, eller ingen, kunskap om den inre systemdynamiken. Vi kan inte säga något om de sätt med vilka de olika faktorerna påverkar ett utfall eller vilka interna relationer som existerar mellan faktorerna, det vill säga hur de samverkar och vilken effekt detta har på utfallet. Genom att uppnå en förståelse för vad, och på vilket sätt som effekterna påverkar utfallet kan vi tillhandahålla bättre, och mer exakta, prediktioner än vad vi annars skulle kunna åstadkomma. En process som utvecklats för att hantera den här sortens problem är data farming – en process som egentligen kan sägas vara en kombination av redan befintliga processer och tekniker som tillsammans utgör ett verktyg att använda för att maximera den information som finns tillgänglig. Fokus ligger på att försöka framställa ett så komplett landskap över potentiella utfall som möjligt och urskilja områden av speciell betydelse, snarare än att ringa in ett enskilt svar. Förutom att identifiera viktiga effekter och relationer mellan faktorer i en simuleringsmodell så läggs även stor vikt vid att fånga upp eventuella anomalier och inkludera dessa i beslutsunderlaget. Processen för data farming delas in i sex domäner, vilka inte ska ses som helt fristående processer utan vilka överlappar och är beroende av varandra för att till fullo kunna utnyttja processens styrkor och kunna tillhandahålla en användbar slutprodukt. Dessa delområden är *modellutveckling*, *snabb framställning av scenarioprototyper*, *experimentdesign*, *högprestandaberäkningar*, *dataanalys och visualisering*, och *samverkande processer*. I denna rapport fokuserar vi omvärldsanalysen på experimentdesign och analys och visualisering av simuleringsutdata.

Nyckelord: Data farming, simulering, experimentdesign, försöksplanering, dataanalys, visualisering.

Summary

How can we support decision makers when the only thing that is absolutely certain is that nothing is completely certain? For each new task we face, we probably know about the factors of influence, but we usually have very little or no knowledge of the internal system dynamics. We cannot say anything about the way in which the various factors have impact on an outcome, or which internal relationships that exist between the factors, i.e., how they interact and what affect this has on the outcome. By gaining an understanding of what impact the effects have on the outcome, we can provide better and more accurate predictions than we could otherwise achieve. A process developed to handle this kind of problem is data farming – a process that can be said to be a combination of already existing processes and technologies that make up a tool suite to maximize the information available. The focus is on trying to produce a sufficiently complete landscape of potential outcomes, and identify areas of special significance, rather than identifying an individual response. In addition to identifying significant effects and relationships between the factors, great importance is also placed on detecting possible anomalies and include them in the decisions. The data farming process is divided into six domains, which should not be seen as completely independent processes, but overlap and are interrelated to fully exploit the process's strengths and to provide a usable end product. These areas include *model development*, *rapid prototyping of scenarios*, *design of experiments*, *high performance computing*, *data analysis and visualization* and *collaborative processes*. In this report we focus the survey on design of experiments and analysis and visualization of simulation data.

Keywords: Data farming, simulation, design of experiments, data analysis, visualization.

Innehållsförteckning

1 Inledning

De frågeställningar som dagens militära beslutsfattare ställs inför är ofta av mer komplex karaktär än någonsin tidigare och krigföringen är långt mer irreguljär än vad konventionen föreskriver. Följaktligen måste vårt sätt att angripa problem anpassas efter de nya spelreglerna så att beslutsfattarna kan rustas för att bemöta dessa nya omständigheter.

1.1 Nya omständigheter – nya utmaningar

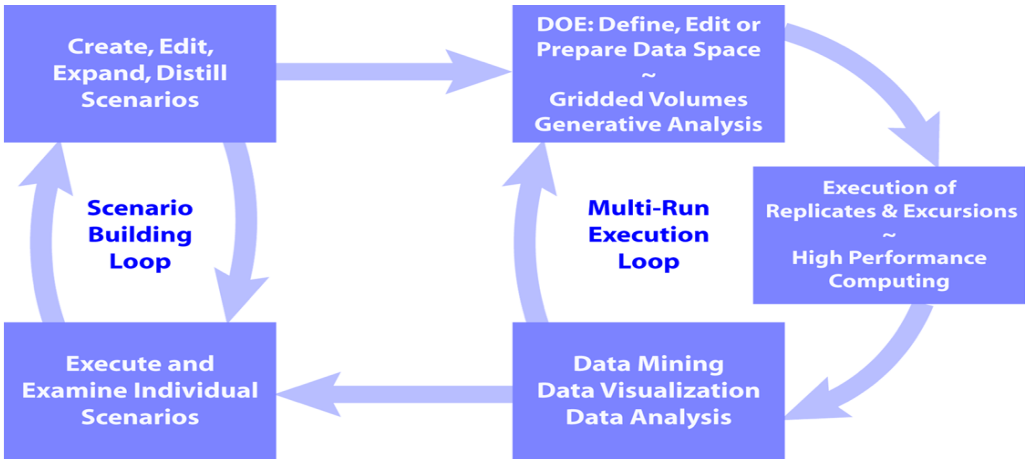
Hur kan vi då stödja beslutsfattare när det enda som är riktigt säkert är att ingenting är helt säkert? För varje ny uppgift vi ställs inför har vi förmodligen kunskap om vilka faktorer som påverkar, men däremot har vi oftast väldigt liten, eller ingen, kunskap om den inre systemdynamiken. Vi kan inte säga något om de sätt med vilka de olika faktorerna påverkar ett utfall eller vilka interna relationer som existerar mellan faktorerna, det vill säga hur de samverkar och vilken effekt detta har på utfallet. Genom att uppnå en förståelse för vad, och på vilket sätt som effekterna påverkar utfallet kan vi tillhandahålla bättre, och mer exakta, prediktioner än vad vi annars skulle kunna åstadkomma.

Det existerar dock ett nästintill oändligt landskap av möjligheter, vilket gör det väldigt svårt, om inte direkt omöjligt, att med vanliga metoder kunna överblicka landskapet på ett användbart sätt. Enligt Pareto-principen står endast en mindre mängd indata för den största systempåverkan, och således har inte alla faktorer lika stor betydelse. Följaktligen vill vi studera landskapet för att kunna fokusera på just de faktorer som är av speciell vikt och lära oss mer om deras påverkan. Själva kärnan i vår uppgift är med andra ord att skaffa oss en klarare bild över vilka faktorer vi skall fokusera på inom ramen för uppgiften samt hur dessa indata förvandlar och förändrar utdata.

Förutom att få en överblick över den stora mängden möjligheter blir uppgiften än mer komplicerad då problemen också ofta inkluderar ett stort antal olika faktorer som kan anta många olika värden. Andra problemkaraktäristika är exempelvis heterogena fel, icke-linjära systemsvar, ett flertal signifikanta effekter samt många komplexa beroenderelationer mellan faktorerna.

1.2 Data Farming

En process som utvecklats för att hantera den här sortens problem är data farming – en process som egentligen kan sägas vara en kombination av redan befintliga processer och tekniker som tillsammans utgör ett verktyg att använda för att maximera den information som finns tillgänglig. Data farming syftar till att skapa insikter till problemformuleringar och är en iterativ process bestående av en ”slinga av loopar”, se figur 1.



Figur 1. Scenarioutvecklingsloop och experimentloop [1].

Fokus ligger på att försöka framställa ett så komplett landskap över potentiella utfall som möjligt och urskilja områden av speciell betydelse, snarare än att ringa in ett enskilt svar. Förutom att identifiera viktiga effekter och relationer mellan faktorer i en simuleringsmodell så läggs även stor vikt vid att fånga upp eventuella anomalier och inkludera dessa i beslutsunderlaget för att på så sätt undvika att beslutsfattaren riskerar att överraskas av det oväntade. Det finns kanske inget som kan sägas vara ett optimalt beslut i ett system där det existerar motståndare som agerar efter eget huvud, men genom att beslutsfattaren tillåts att förstå landskapet av möjligheter är tanken att mer välunderbyggda beslut kan fattas.

Utifrån kännetecknen hos de problem som skall lösas finns ett behov av att på ett funktionellt sätt kunna modellera olinjäriteter, abstraktioner och påverkan mellan modellernas olika delar. Det är kombinationen av enkla, effektiva och abstrakta modeller, samt högprestandaberäkningar som tillsammans med effektiv experimentdesign möjliggör snabb utforskning av ett utfallsrum. Enkla modeller underlättar att hantera ett stort antal körningar, vilket i sin tur möjliggör att en stor parameter- och värderymd behandlas och att utfallsrummet utforskas. Resultatet blir således ett landskap av utdata som sedan kan användas för att analysera trender, upptäcka anomalier och för att skapa insikter om multipla parameterdimensioner. Efter att datarymden initialt utforskats, ”odlas” ny data genom att expandera eller öka upplösningen hos initialvillkor och parametrar, varefter processen fortsätter iterativt.

Under data farming-processen möjliggör vunna insikter att odling av data kan koncentreras till intressanta områden, samt genom att viktiga parametrar och komponenter läggs till. Detta illustrerar processens iterativa natur – data odlas och utvinns inte enbart, utan man återvänder och odlar ytterligare data i områden av intresse i syfte att på ett mer adekvat sätt adressera frågan. Denna process fortgår till dess att insikter relevanta för beslutsfattarna har uppnåtts. Förutom att identifiera landskapets generella karaktärsdrag strävar analysen även efter att skänka förståelse för exempelvis resultatens spridning och centraltendenser samt faktorernas inbördes förhållanden och tröskelvärden.

Data farming-metodik tillämpar en simuleringsbaserad holistisk och iterativ metod för att analysera komplexa system. Utmaningen för alla simuleringssystem som kör endast *en* simulering är att det bara ger resultat om *en* viss situation och dess omständigheter. Det ger således inga slutsatser om olika förhållanden – inklusive identifiering av bästa och värsta scenarier.

Data farming är en metodik för att ge svar på dessa frågor. Det är en simuleringsbaserad analytisk process som:

- är tillämpligt för kvantitativ analys av komplicerade frågor,
- möjliggör ”what if”-analyser,
- ger robusta resultat,
- jämför resultat baserade på olika kategorier.

Kärnan i data farming bygger på att en rik och skiftande mängd av olika simuleringskörningar genomförs på superdatorer för att kontrollera olika antaganden, för att få nya insikter i relevanta relationer, samt för att få mer robusta utsagor om möjligheter och risker i specifika uppdragssituationer. Detta uppnår man genom att systematiskt variera olika parametervärden för de beslutade indataparametrarna som antas vara avgörande som ett mått på effektivt.

1.3 De sex domänerna

Processen för data farming delas in i sex domäner, vilka inte ska ses som helt fristående processer utan vilka överlappar och är beroende av varandra för att till fullo kunna utnyttja processens styrkor och kunna tillhandahålla en användbar slutprodukt.

Dessa delområden är *modellutveckling*, *snabb framställning av scenarioprototyper*, *experimentdesign, högprestandaberäkningar*, *dataanalys och visualisering*, och *samverkande processer*. Experimentdesign och dataanalys och visualisering är två delar av data farming som är sammanflätade på ett sådant sätt att det är lämpligt att presentera dem tillsammans.

Experimentdesign hjälper till att övervinna problemets dimensionalitet. Jämfört med en brute force-strategi för att köra simuleringsexperiment, ger det effektivare sätt att sätta upp experimentet på ett sätt som underlättar uppföljning med dataanalys och visualisering av resultat inom en rimlig tid. Den typ av experimentdesign som används i ett simuleringsexperiment dikterar vilken utdata som kommer att genereras och samlas in från simuleringen. Det påverkar också vilka metoder som kan användas i analysen av utdata från simulering.

Visualiseringen består av att analysera simuleringens utdata med lämpliga tekniker samt presentera resultaten till beslutsfattare. Även med en smart experimentdesign så skapar simuleringsexperiment stora mängder flerdimensionella data som kräver sofistikerad dataanalys och visualiseringstekniker.

I denna rapport fokuserar vi omvärldsanalysen på *experimentdesign* (kapitel 2) och *analys och visualisering av simuleringsutdata* (kapitel 3).

1.3.1 Modellutveckling

Inom data farming benämns en modell som en destillations modell eftersom man, för att möta problemet, reducerar frågeställningen till en sådan enkel representation som möjligt. På så sätt minskas detaljeringsnivån utan att frågeställningens kärna går förlorad. Destillationen kan säga vara intuitiv, flyttbar och odlingsbar och ska även uppfylla de höga modellkrav som ställs, exempelvis skall de kunna hantera olinjäritet, anpassning, abstraktioner och påverkan mellan modellens olika delar för att bara nämna några saker som uppkommer i den verklighet som vi försöker efterlikna.

Inom data farming är destillationerna ofta agentbaserade, vilket inte är ett krav men eftersom agentmodeller lämpar sig väl för att ta fram prototyper används de ofta. I dessa destillationer försöker man modellera de kritiska faktorerna som är av intresse utan att explicit modellera alla de fysikaliska detaljerna. Några exempel på system för agentbaserade destillationer som ofta återkommer inom data farming är *Pythagoras* och *MANA*. Dessa kan med lätthet köras många gånger för att testa en mängd olika parametervärden och möjliggöra en bredare inblick i landskapet av möjligheter.

En simuleringsmodell för data farming används i en konstruktiv simulering. Ett krav på modellen är att den måste startas automatiskt utan behov av någon interaktion med användaren. Dessutom måste den också kunna avslutas automatiskt så snart simuleringskörningen är klar. För att kunna starta körningen av modellen automatiskt måste det också vara möjligt att automatiskt ange indata till simuleringssystemet. Simuleringsmodellen måste kunna återge specifika simuleringskörningar med identiskt resultat (t.ex. för att analysera extremvärden) – själva modellen kan dock vara stokastisk. Befintliga modeller kan anpassas med hänsyn till dessa krav för att användas i data farming-experiment. En möjlig lösning är att konvertera gränssnitt och beslutsfaktorer i gamla simuleringssystem och modeller till beslutsvariabler i ett modellgränssnitt för data farming.

1.3.2 Högprestandaberäkningar

De mängder data som måste behandlas är oftast av sådana kvantiteter att högprestandaberäkningar är nödvändiga. Den enorma mängd utdata som körningarna genererar samlas upp i verktyg designade för detta ändamål. Förutom att modellerna körs med olika värden och parametrar så körs även samma set av parametrar med olika frövärden för att få en förståelse för resultatens fördelning över samma parameterset.

1.3.3 Analys och visualisering av simuleringsutdata

Eftersom högprestandaberäkningar just genererar så pass stora mängder utdata så blir även verktyg och metoder för visualisering av stor vikt. Utdata måste visualiseras för att underlätta det fortsatta analysarbetet med det producerade underlaget. Målet som eftersträvas är att sammanfoga alla små bitar av information till en helhet som ger beslutsfattaren förståelse. Genom analys vill man undersöka spridning, centraltendens och eventuella anomalier samt fördelningens utseende och form för vilket man ofta använder statistisk sammanfattning, histogram och boxplottar.

Utöver statistisk sammanfattning är några vanliga analysmetoder exempelvis regression, träd och datavisualisering. För datavisualisering använder man sig ofta av interaktionsprofiler, parvisa plottar och träd.

1.3.4 Snabb framställning av scenarioprototyper

En snabb framställning av scenarioprototyper förbättrar data farming-processen än mer, genom att den möjliggör den iterativa naturen hos användningen av högprestanda-beräkningar för att köra modeller många gånger över olika initialvillkor i syfte att ge en förståelse för möjliga anomalier, trender och resultatfördelning.

Då scenarioprototyper framställs är eftersträvan inte att framställa den allra mest exakta eller detaljerade scenariot utan man ämnar snabbt uppnå ett scenario som är bra nog. Under detta iterativa arbete plockas delar bort så fort de anses överflödiga och arbetet fortsätter och fler egenskaper läggs till.

1.3.5 Experimentdesign

Data farming manipulerar simuleringsmodeller till sin fördel genom experimentdesign eftersom simulering av vartenda värde och kombination av värden inte är möjligt – högprestandaberäkningar till trots. Valet av experimentdesign begränsar den information som kan extraheras från modellen vilket understryker vikten av att fylla upp parameterrummet så effektivt som möjligt.

Latinska hyperkuber är en experimentdesign som fokuserar på många faktorer med komplexa modeller och då dessa även kan kombineras med mer klassisk experimentdesign för att skräddarsy designen utifrån problemet i fråga brukar de ofta användas inom data farming. Fokus ligger även på sekventiella metodiker, vilket innebär att man, efter initialt arbete med utforskning av landskapet, skär bort de möjligheter som bedöms vara av mindre intresse, och sedan fortsätter att utforska den del av landskapet som återstår.

Ytterligare en styrka med att använda latinska hyperkuber är att designen fungerar speciellt väl då faktorerna är kvantitativa och förkunskapen gällande dessas respons är låg. Några andra styrkor är också dess effektivitet, rymdfyllande egenskaper, designflexibilitet och analysflexibilitet. Ortogonala och nästan ortogonala latinska hyperkuber har fördelar när det gäller att matcha modellen till data. Genom att använda *nästan ortogonala latinska hyperkuber* (NOLH) garanteras att modellens faktorer är okopplade. Med denna experimentdesign uppnås en utforskning utan några större hålrum och man kan på detta sätt även identifiera dominanta faktorer och tillgodose icke-linjära beteenden.

1.3.6 Samverkande processer

Data farming-processens sjätte och sista domän är en sammanhållande del vars syfte är att integrera de övriga domänernas samarbete över domängränserna. För att kunna nyttja data farming:s styrkor till fullo måste samverkan ske på många plan, och delprocessen syftar också till att möjliggöra samarbete mellan människor så att de kan utbyta erfarenheter och bidra till ett effektivare arbete.

1.4 Den iterativa processen

Den samlade data farming-processen skulle översiktligt kunna beskrivas i fyra faser:

- Scenarioskapande – den modell som ska spegla frågeställningen utvecklas och finslipas, och processen upprepas (ibland är det till och med nödvändigt att gå tillbaks till själva grundfrågan och även justera den) till dess att man kommit fram till en godtagbar modell. Resultatet blir ett slags basfall som kommande fas utgår från.
- Scenariostyrd rymdexekvering – de inparametrar som ska studeras och vilka processer som skall användas för att variera dem bestäms i en studie. Utifrån basfallet undersöker man också vilka möjliga avvikelser som kan göras gällande scenariots initialvillkor. Resultatet blir en studie vilken har stor betydelse under nästa fas.
- Högprestandaberäkningar – studien används i denna fas för att styra genomförandet av många körningar i miljön för högprestandaberäkningar. Varje körning producerar utdata som samlas upp av data farming-systemet.
- Analys – simuleringsresultaten som genererades i föregående fas analyseras, och beroende på vilka slutsatser som dras kan nästa steg bli att återgå till fas ett eller fas två och börja om processen därifrån.

1.5 Data Farming i praktiken

Data farming kan användas på många olika sätt för att stödja modellering och beslutsstöd. Till exempel för:

- Känslighetsstudier – modeller med olika komplexitetsgrad är föremål för icke-linjära beteenden som kan variera över modellens indatarymd. Med data farming ges vi möjligheten att undersöka stora indataområden med hög upplösning av parameterrymden för att undersöka modellens statistiska variation.
- Validering och verifiering – data farming ger modellbyggare en möjlighet att fullt ut testa en modells uppträdande mot olika indata över ett brett område med möjliga och potentiellt oförutsedda kombinationer av indataparametrar. Vi kan granska resultaten för att fastställa om modellen algoritmer utförs korrekt och för att jämföra resultaten med verkligheten.
- Modellutveckling och spel – alla modeller är förenklingar av verkligheten. För att finslipa modellen och dess parametrar till att bättre representera verkligheten, körs modellerna ofta upprepade gånger för att ”styra” parametrarna. Dessutom kräver arbetet med att utveckla modellen otaliga körningar som hjälp vid felsökning och algoritmutveckling. Möjligheten att köra modeller över en stor parameterrymd snabbar också upp utvecklingsprocessen.
- Scenarioanalys – när modeller väl är utvecklade ska de exekveras. Resultaten av körningarna studeras för att ge insikt eller för att ta upp verkliga frågeställningar. Data farming låter modellen köras över ett större antal indataparametrar och ett större antal slumpmässiga variationer, som kan ge beslutsfattarna en mer fullständig bild av möjliga utfall och systemets dynamik.
- Trender och extremvärden – traditionellt körs modeller några få gånger för att göra en scenarioanalys som bygger på ett litet antal möjliga utfall. Någon få sammanfattande statistiska samband genereras för att representera resultaten. Om man däremot undersöker ett bredare parameterutrymme kan trender och relationer mellan indata och effektivitetsmått studeras. Lika viktig är förmågan att identifiera vilka parameterkombinationer eller slumpmässiga variationer som

leder till extremvärden, dvs. särskilda fall som kan tyda på modellproblem, eller delar av parameterrummet med hög risk eller hög möjlighet.

- Heuristisk sökning och upptäckt – data farming omfattar förmåga att tillämpa iterativa metoder för modellanalys som genetiska algoritmer och andra avancerade optimerings- och sökmetoder.
- Generering av massiva testdatauppsättningar – data farming kan användas tillsammans med modeller för att generera stora testdatamängder för att testa maskininlärningsalgoritmer och andra data mining-verktyg. Detta är särskilt värdefullt när faktiska data inte är tillgängliga av säkerhetsskäl eller när sekretess råder.

2 Experimentdesign

Experimentdesign är en teknik för informationsinsamlings som innebär att vi gör experiment genom att definiera faktorer med varierande nivåer. Försvarsrelaterade simuleringsmodeller har vanligen ett stort antal faktorer som används för att undersöka intressanta åtgärder via icke-linjära modeller med heterogena fel. Dessa simuleringsmodeller kan ha högre ordningens interaktioner vars fel inte nödvändigtvis är normalfördelade. Simuleringsmodellerna utgör inte en ”black box” då modelleringen kräver kompetens av domänexperter.

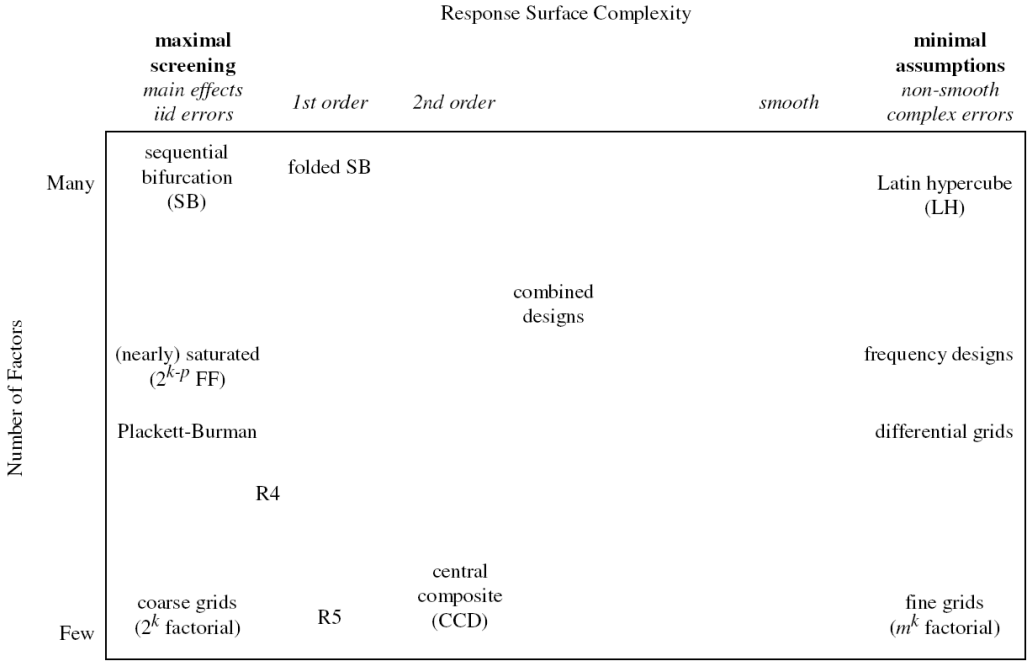
Teknologiska framsteg inom högpresterande databehandling har lett datorutvecklingen mot datorer som idag kan utföra drygt 10^{16} operationer per sekund [2]. Om en simuleringsmodell som har 100 faktorer var och en med 2 parametervärden utförs med *en* enda operation på en sådan dator kommer simuleringen att terminera först efter 4 miljoner år. Hög dimensionalitet är således ett problem vid simulering. Det gör det nödvändigt med metodik som går bortom högprestandaberäkningar och en enkel brute-force approach. Experimentdesign kan användas för att kringgå dimensionalitetsproblemet genom att tillhandahålla representativa stickprov över hela indatarummet och därmed minska de beräkningar som krävs för att nå slutsatser inom rimlig beräkningstid.

Klassisk försöksplanering[1] går tillbaka till 1920-talet och idéer hos Sir John Russell och Ronald A. Fisher. Russell publicerade 1926 en artikel med titeln ”Field experiments: How they are made and what they are” [3] vilken utgjorde den tidens state-of-the-art inom området. Fisher första bidrag kom som ett svar till denna artikel senare samma år. Den hade titeln ”The arrangement of filed experiments” [4]. Användningen av experimentdesign för simuleringsexperiment är dock av mycket senare datum. Under 1970-talet publicerades arbeten som i huvudsak anpassade metoder från klassiska fältförsök till simuleringsvärlden. Inte förrän på 1980-talet utvecklades många nya metoder direkt avsedda för simuleringsexperiment. Vi ger i detta kapitel en översikt över ett antal designmetoder för utformning av simuleringsexperiment utan krav på att vara heltäckande. I denna översikt listar vi alternativa metoder för experimentdesign utifrån simuleringsmodellens antal faktorer och dess komplexitet.

Vi försöker sammanfatta de frågeställningar som är av betydelse vid val av en experimentdesign för att samla in och analysera utdata från körningen av en simuleringsmodell. Dessa frågeställningar kan vara, vilken typ av fråga försöker vi besvara med exekvering av simuleringsmodellen, vilket slags beteende har simuleringsmodellen, vilka gränsvärden finns för olika faktorer i modellen, vilken slags efterföljande dataanalys och visualisering avser vi genomföra, och hur ska resultaten presenteras. Allt detta påverkar hur vi väljer att sätta upp experimentdesignen.

I figur 2 beskriver vi olika metoder för experimentdesign utifrån två olika dimensioner. Den vågräta axeln representerar modellens komplexitet från enkla till komplexa svarsfunktioner. Längs axeln presenterar vi olika metoder för experimentdesign som passar till olika problemtyper av varierande komplexitet. Den vertikala axeln representerar antal faktorer i modellen. Således representerar det nedre vänstra hörnet enkla svarsfunktioner med en handfull faktorer. Övre högra hörnet representerar mycket komplexa svarsfunktioner med många faktorer. Det som presenteras är inte en komplett lista över alla tillgängliga designmetoder, utan en beskrivning av dem som verkar mest lovande och är lättillgänglig eller ganska lätt att generera.

[1] För simuleringsbaserad försöksplanering väljer vi istället att översätta *design of experiments* med experimentdesign.



Figur 2. Rekommenderad experimentdesign är beroende på antal faktorer i simuleringsmodellen samt antagande om modellens komplexitet [5].

En grundläggande princip är att undvika att börja experiment genom att fokusera på ett litet antal faktorer. I stället bör man använda experimentdesigner med många faktorer. På detta sätt kan analytiker överväga betydelsen hos ett stort antal faktorer i simuleringen. Analytikern som önskar göra förenklande antaganden kan börja i vänster sidan av figuren, vilket tenderar att minska storleken på den initiala utdatan från simulatorn. Antaganden som införs bör kontrolleras senare att de är giltiga. Alternativt kan analytikern starta från det övre högra hörnet för ett inledande experiment om lite är känt om vilken typ av svarsfunktion som kan väntas.

Om detta inledande experiment inte helt behandlar det huvudsakliga målet, så kan preliminära resultat användas för att utforma nya experiment som fokusera på de faktorer eller regioner som verkar mest intressanta. Detta motsvarar att flytta nedåt i figur 2 för att fokusera på de faktorer som valts ut efter det inledande experiment samtidigt som resterande faktorer endast tillåts ta ett reducerat antal värden. Om antaganden om svarsfunktionens komplexitet görs redan för det första försöket (genom att röra sig mot nedre högra hörnet), så bör dessa antaganden giltighet kontrolleras. Om få antaganden görs initialt och den inledande analysen tyder på att svarsfunktionen inte är speciellt komplicerad, så kan analytiker dra fördel av högeffektiva metoder i nedre vänstra hörnet i efterföljande experiment.

Oavsett vilken experimentdesign som används för simuleringen så bör man efter det att simuleringen är genomförd kontrollera att gjorda antaganden faktiskt håller. Har man använt metodik från högra sidan av figur 2 så har man gjort få antagande om modellens karakteristik. Om man å andra sidan startar i övre vänstra hörnet (*sequential bifurcation*) är man antagligen ute för att identifiera ett fåtal faktorer av avgörande betydelse för ett efterföljande simuleringsexperiment. Det medför att antalet antaganden man då gör i det efterföljande experimentet reduceras tillsammans med reduceringen av antalet faktorer och risken för felaktiga antaganden minskar. Om man däremot böjar experimentet med modeller från nedre vänstra hörnet i figuren då är det extra viktigt att verifiera att gjorda antaganden verkligen håller.

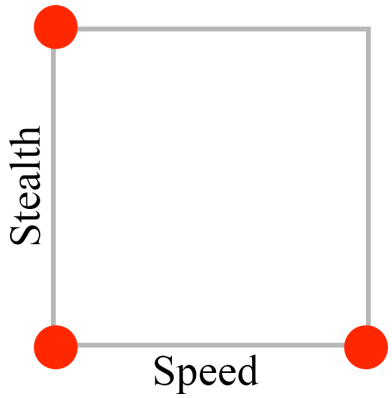
Några möjliga mål med simuleringsexperiment kan vara att verkligen försöka förstå modellens övergripande beteende, och att hitta gränserna för de indata som leder till robust beteende hos modellen. Detta kan vara viktigare än en mer traditionell studie av olika

faktorers påverkan på modellens utdata, samt att använda modellen för att försöka göra prediktioner. Man bör också överväga om det är lämpligt att genomföra simulerings-experiment som en serie sekventiella experiment. Gör man det, så kan man vinna initiala insikter som kan användas för att styra efterföljande experiment, t.ex. vilka faktorer som bör ingå i experimentet och vilka parameterområden som bör undersökas ytterligare, t.ex. med förfinad upplösning.

2.1 Metoder

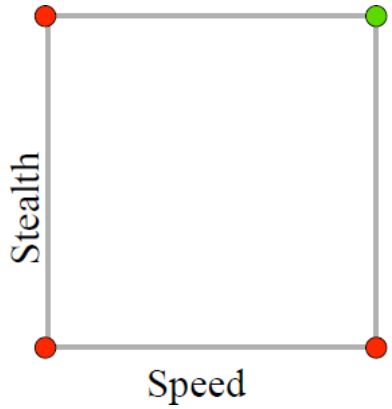
2.1.1 Faktordesigns

En strategi som kan vara problematisk uppstår när man börjar med ett huvudscenario och variera endast en faktor i taget. Låt oss överväga ett problem med två faktorer hastighet och smygförmåga (eng. stealth) där initialtillståndet motsvarar låg hastighet och låg smygförmåga. Varierar man varje faktor var för sig till dess högre värde får man resultatet i figur 3. Det verkar som om ingendera faktorn är viktig.



Figur 3. Sampling av indata med en faktor åt gången [6].

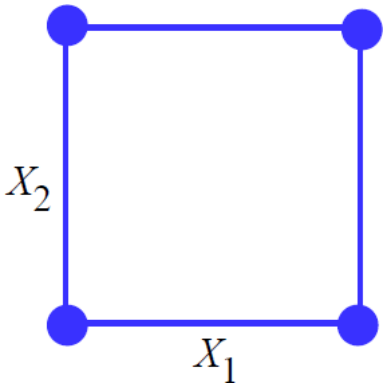
I figur 4 har vi en 2^2-faktordesign för samma tankeexperiment där vi går igenom alla kombinationer av faktorvärden som indata till simuleringen. För 2^2-designen är dock allt som kan sägas att när båda faktorerna tar sitt högra värde så är experimentet framgångsrikt. Vi återkommer till detta experiment.



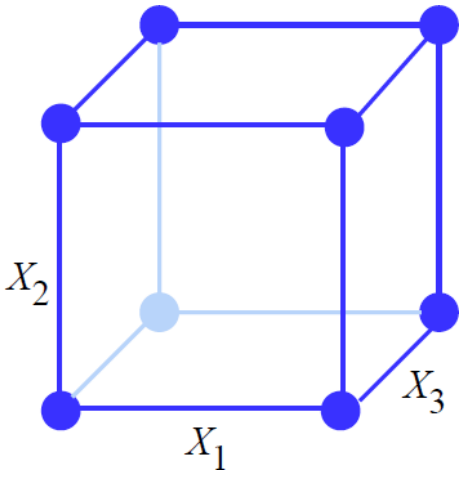
Figur 4. En 2^2-faktordesign undersöker alla kombinationer av låga och höga värden [6].

2.1.1.1 Problembeskrivning

Om man har ett fåtal faktorer är de möjligt att göra en komplett faktoranalys [6]. Metoden passar alla problem där beräkningstiden för summan av alla simuleringar är hanterbar. Den enklaste faktordesignen (eng. factorial design) är en 2^k-design för k stycken faktorer, figur 5 och 6. Här modellerar man två alternativa nivåer för varje faktor. Denna design passar de simuleringsmodeller som förutom att endast ha ett fåtal faktorer även har linjära responsfunktioner.



Figur 5. En 2^2-faktordesign för två faktorer med två nivåer per faktor [6].

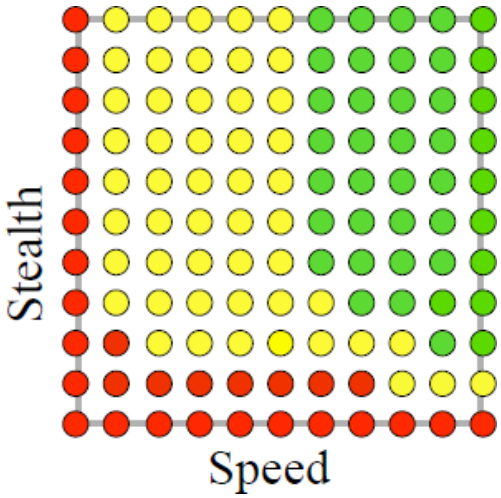


Figur 6. En 2^3-faktordesign för tre faktorer med två nivåer per faktor [6].

Det går även att modellera kombinationer av faktorer som om de vore enskilda faktorer, man kan därmed även fånga interaktioner mellan faktorer. Man kan göra det för par, tripplar, etc., antingen för alla kombinationer mellan faktorernas nivåer eller för ett utvalt antal kombinerade faktorer om man har hunnit lära sig tillräckligt om modellens beteende för att kunna förutspå vilka kombinationer av faktorer som interagerar på ett ickelinjärt sätt.

Om man undersöker varje faktor på endast två nivåer innebär det att vi inte har någon aning om hur simuleringen beter sig för olika faktorkombinationer i det inre av den experimentella regionen. Har man en mer komplex simuleringsmodell så kräver det en finare upplösning på indata för varje faktor för att avslöja komplexiteten i landskapet. Man kan t.ex. komplettera med en mittennivå för varje faktor och får då en 3^k-design. Modellen blir nu mer robust men dess beräkningskomplexitet ökar. Vid t.ex. 15 faktorer ökar dock beräkningstiden med drygt 400 gånger vid övergång från en 2^k- till 3^k-design. Generellt kan man ha m nivåer för varje faktor för en m^k-design. Vi rör oss då från vänster till höger i den nedre delen av figur 2.

Låt oss för ett ögonblick återvända till resultatet av vårt tankeexperiment med hastighet och smygförmåga. I figur 7 har vi en 11^2-design. Mer information förmedlas av 11^2-faktordesignen. Här ser vi att om en viss nivå på smygförmågan kan uppnås så är hastigheten avgörande. I figurerna representera gröna punkter goda resultat, de gula punkterna resultat av mellanliggande kvalité och de röda punkterna på den vänstra sidan och längs botten dåliga resultat.



Figur 7. En 11^2-faktordesign finner hela utfallsrummet till en högre beräkningskostnad [6].

2.1.1.2 Metodbeskrivning

Med en 2^k-design för k stycken faktorer simuleras varje faktor med två alternativa nivåer, t.ex. högt respektive lågt värde för varje parameter (oftast de två gränsvärdena för varje faktor). Man simulerar då alla 2^k resulterade kombinationer.

När man arbetar med kodade nivåer (t.ex. −1, +1), hittas interaktionskolumnerna genom att multipliceras kolumnerna för de associerade huvudsakliga effekterna, se tabell 1 för en 2^3-faktordesign. När varje faktor har tre nivåer, är konventionen att använda −1, 0 och +1 för kodade nivåer.

Indata-punkt	X_1	X_2	X_3	X_1X_2	X_1X_3	X_2X_3	$X_1X_2X_3$
1	–1	–1	–1	+1	+1	+1	– 1
2	+1	–1	–1	–1	–1	+1	+1
3	–1	+1	–1	–1	+1	–1	+1
4	+1	+1	–1	+1	–1	–1	–1
5	–1	–1	+1	+1	–1	–1	+1
6	+1	–1	+1	–1	+1	–1	–1
7	–1	+1	+1	–1	–1	+1	–1
8	+1	+1	+1	+1	+1	+1	+1

Tabell 1. Termerna för en 2^3-faktordesign med åtta indatapunkter och sju olika termer (kolumner) [6].

I tabell 1 finns det sju olika termer (tre huvudsakliga effekter, tre tvåvägsinteraktioner och en trevägsinteraktion) som vi uppskatta från ett 2^3-faktorsexperiment.

Det är också möjligt att bygga en hierarkisk modell där man skapar faktorer av faktorer genom att gruppera faktorerna och endast samtidigt variera faktorernas nivåer inom en och samma grupp. Beräkningskomplexiteten begränsas då till 2^l där l är antalet grupper. Man kan sedan göra nya experiment där man löser upp en grupp och förfinar analysen.

Tyvärr är faktordesign behäftat med exponentiell beräkningskomplexitet, men dessa designs är enkla att använda då antalet faktorer är lågt och kan ge en detaljerad översikt över utfallsrummet så länge beräkningstiderna är rimliga.

2.1.2 Fraktionala faktordesigns

2.1.2.1 Problembeskrivning

Om vi är villiga att anta att vissa högre ordningens interaktioner inte är viktigt, kan vi minska antalet simulerings körningar som krävs genom att reducera antalet indatapunkter till simuleringen med en fraktional faktordesign.

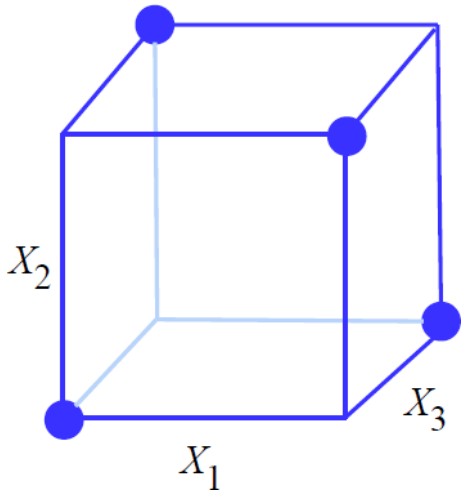
2.1.2.2 Metodbeskrivning

Vi illustrerar metodiken med hjälp av en 2^k-faktordeseign. Vi utgår från tabell 1 som visar alla interaktioner för ett 2^3-faktorförsök och ersätter varje interaktionsterm med en ny faktor. Den resulterande designen kallas en 2^{7-4}-fraktional faktordesign eftersom designen varierar sju faktorer i bara $2^{7-4} = 8$ indatapunkter istället för $2^7 = 128$ indatapunkter i en traditionell 2^7-faktordesign. Det ger en reducering av beräkningstiden med en faktor 16. Faktorn X_4 använder den kolumn som skulle motsvara en X_1X_2 interaktion, faktorn X_5 kolumnen som skulle motsvara en X_1X_3 interaktion, osv. för X_6 och X_7, se tabell 2.

Indata-punkt	X_1	X_2	X_3	X_4	X_5	X_6	X_7
1	–1	–1	–1	+1	+1	+1	–1
2	+1	–1	–1	–1	–1	+1	+1
3	–1	+1	–1	–1	+1	–1	+1
4	+1	+1	–1	+1	–1	–1	–1
5	–1	–1	+1	+1	–1	–1	+1
6	+1	–1	+1	–1	+1	–1	–1
7	–1	+1	+1	–1	–1	+1	–1
8	+1	+1	+1	+1	+1	+1	+1

Tabell 2. Termerna för en 2^{7-4}-fraktional faktordesign med åtta indatapunkter och sju olika termer (kolumner) [6].

I figur 8 visas indatapunkterna för en 2^{3-1} fraktional faktordesign för att undersöka tre faktorer, var och en på två nivåer med bara $2^{3-1} = 4$ simuleringar. Notera som exempel, att det finns två punkter på vart och ett av vänster och höger sida på kuben, och var och en av dessa sidor har en instans av X_2 på varje nivå och en instans av X_3 på varje nivå, varför vi kan isolera effekten för faktor X_1, osv.



Figur 8. En 2^{3-1} fraktional faktordesign [6].

Upplösning i en design är avgörande för vilken komplexitet som kan hanteras i simuleringsmodellen. Har vi en simuleringsmodell med komplex responsfunktion där interaktion mellan olika faktorer spelar en stor roll för resultatet bör vi välja en design som tillåter att vi kan analysera faktorer utan risk för sammanblandning. Två faktorer sägs vara sammanblandade när de inte kan uppskattas separat.

En designs upplösning (eng. resolution) definieras som *ett* högre än den lägsta graden av interaktioner som sammanblandas med de huvudsakliga effekterna. Som exempel så har en design upplösning 3 (R3) om det finns någon huvudsaklig effekt som är sammanblandad med någon tvåvägsinteraktion. En design sägs ha upplösning 3 (R3) om det är en 2^{k-p}-fraktional faktordesign där $k = 2^n - 1$ där $n \geq 0$. Dessa är enkla att konstruera, se tabell 2.

I allmänhet gäller att för en design med upplösning R är ingen p-faktoreffekt förväxlad med någon annan effekt som innehåller mindre än $R - p$ faktorer [7]. Vi har som exempel (tabell 3),

- en design med upplösning R = III förväxlar *inte* huvudsakliga effekter med varandra (t.ex. X_1 med X_2) men förväxlar huvudsakliga effekter med två-faktor interaktioner (t.ex. X_1 med X_1X_2),
- en design med upplösning R = IV förväxlar *inte* huvudsakliga effekter med två-faktor interaktioner, men förväxlar huvudsakliga effekter med tre-faktor interaktioner och förväxlar två-faktor interaktioner med andra två-faktor interaktioner,
- en design med upplösning R = V förväxlar *inte* huvudsakliga effekter och tre-faktor interaktioner med varandra, men förväxlar huvudsakliga effekter med fyra-faktor interaktioner. Dessutom förväxlas *inte* två-faktor interaktioner med varandra, men förväxlar två-faktor interaktioner med tre-faktor interaktioner.

R3	1-fak	2-fak
1-fak	●	●

R4	1-fak	2-fak	3-fak
1-fak	●	●	●
2-fak	●	●	●

R5	1-fak	2-fak	3-fak	4-fak
1-fak	●	●	●	●
2-fak	●	●	●	●
3-fak	●	●	●	●

Tabell 3. Gröna punkter representerar icke sammanblandade relationer, röda punkter representerar sammanblandade relationer. 1-fak är de huvudsakliga effekterna (t.ex. från X_1), 2-fak är två-faktor interaktioner (t.ex. X_1X_2), etc.

Om interaktioner antas vara närvarande men användarna huvudsakligen är intresserade av att uppskatta första ordningens effekter, då är R4-designs lämpliga. Dessa konstruktioner ger objektiva skattningar av de viktigaste effekterna även om det finns två-faktor interaktioner. R3-designs (som använder alla scenarier för att uppskatta alla effekter) ger mindre medelfel för uppskattning av första ordningens effekter än om man förändrar en faktor i taget mellan lågt och högt värde (dvs. endast använder två scenarier per faktor).

Om det finns starka interaktioner mellan faktorerna är ett alternativ är att använda en R5 fraktional faktordesign vilken tillåter att tvåvägsinteraktioner utforskas (se R5, tabell 3), men kräver färre indatapunkter. Mycket stora R5 fraktionala faktordesigns upp till storlek $2^{120-105}$ kan snabbt generas på mindre än en minut. Dessa tillåter analys av alla huvudsakliga effekter och alla två-faktors interaktioner, se [8] för programkod.

2.1.3 Centralt sammansatta designs (CCD)

2.1.3.1 Problembeskrivning

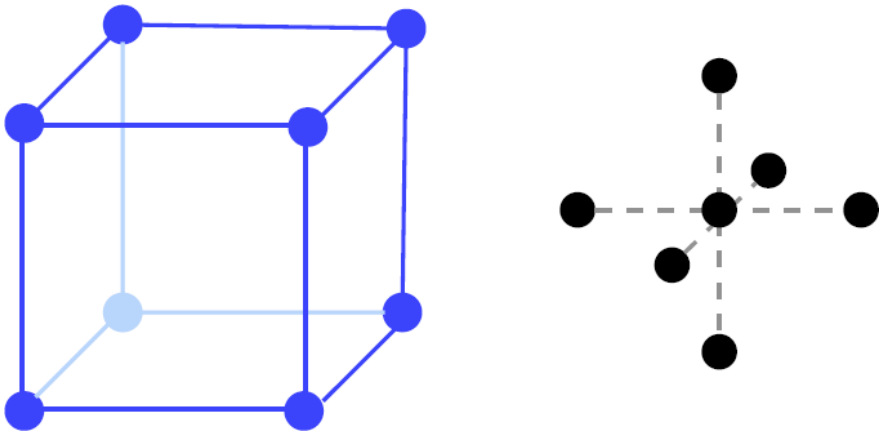
Om vår simuleringsmodell innehåller viktiga interaktionstermer eller kvadratiska termer så kan de inte uppskattas med en faktordesign eller en fraktional faktordesign av ordningen 2^k. Om samtidigt antalet faktorer är alltför stort för att en 3^k-design ska var beräkningstekniskt möjlig att hantera kan en centralt sammansatt design (CCD) (eng. central composite design) vara ett alternativ [5].

2.1.3.2 Metodbeskrivning

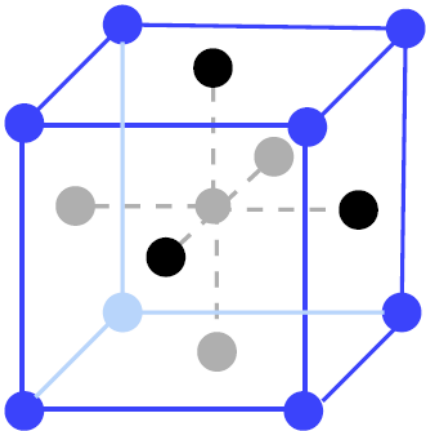
En design som tillåter oss uppskattar alla andra ordningens modeller (dvs. de huvudsakliga effekterna, tvåvägsinteraktioner, och kvadratiska effekter) kallas för en centralt sammansatt konstruktion (CCD). Utgående från en 2^k-faktordesign eller en upplösning 5 (R5) 2^{k-p}-fraktional faktordesign läggs en mittpunkt och två stjärnpunkter till för var och en av faktorerna.

Om man utgår från en fraktional faktordesign eller en R5 design som bas så har en CCD mycket färre indatapunkter en 3^k-faktordesign.

I figur 9 visar vi en 2^3-faktordesign som bas med tillhörande mitt och stjärnpunkter. I figur 10 visas den resulterande ccd:en.



Figur 9. En 2^3-faktordesign som bas med tillhörande mittpunkt och stjärnpunkter [6].



Figur 10. En centralt sammansatt konstruktion (CCD) [6].

2.1.4 Sekventiell bifurkation

2.1.4.1 Problembeskrivning

När antalet faktorer är mycket stort så kan sekventiell screening vara av intresse. En sådan metod är sekventiell bifurkation [9]. Metoden gör antagandet att simuleringen dels kan beskrivas som ett första ordningens polynom, samt att varje faktors påverkan är känt till sitt tecken. Givet dessa förutsättningar kan metoden snabbt eliminera betydelselösa faktorer så att framtida experiment kan fokusera på dem som verkar viktigt.

2.1.4.2 Metodbeskrivning

Metodiken är mycket enkel. Den utgår ifrån att vi kan beskriva simuleringen som en första ordningens polynom där alla faktorers tecken är kända. Vi har responsfunktionen

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_j x_j + \cdots + \beta_K x_K$$

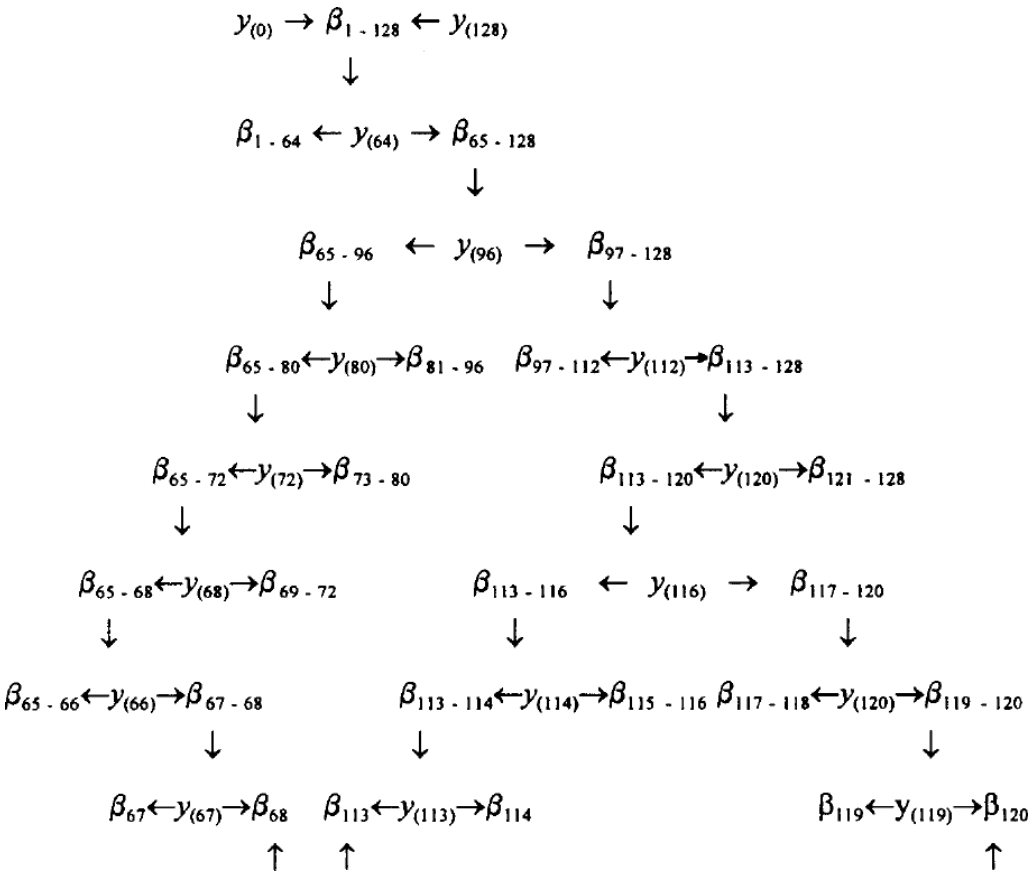
där K är antalet faktorer, x_j tar värdet 0 eller 1 och indikerar om den j:te faktorn är påslagen. Vi antar utan förlust av allmängiltighet att alla faktorernas amplituder $\beta_j \geq 0$. Vi har således en monotont stigande summa. Låt oss kalla

$$y_j = \sum_{i=0}^{j} \beta_i$$

där y_j motsvarar responsfunktionen för en simulering där alla faktorer $x_i = 1$ för $i \leq j$ och alla $x_i = 0$ för $i > j$, dvs. de först j faktorerna är påslagna medan övriga faktorer är avslagna. Då gäller alltid att $y_j \leq y_k$ när $j < k$.

Vi kan nu anta en approach liknande intervallhalvering. Vi börjar med tre simuleringar för y_0 (alla faktorer är avslagna; $x_i = 0$), y_n (de n första faktorerna är påslagna; $x_i = 1$, $i \leq n$, övriga faktorer är avslagna; $x_i = 0$, $i > n$) och y_K (alla faktorer är påslagna; $x_i = 1$), där x_n ligger någonstans mitt emellan x_1 och x_K (för att uppnå maximal effektivitet bör man välja det största n till x_n där $2^n < K$).

Vi kan nu betrakta de två differenserna $y_n - y_0$ och $y_K - y_n$ mellan responserna från de tre genomförda simuleringarna. Den största påverkan från viktiga faktorer finns i intervallet med den högsta differensen. Vi fokuserar på detta intervall och fortsätter med intervallhalveringen, osv. I figur 11 visas ett exempel på hur man med denna process finner de $k = 3$ viktigaste faktorerna i en simuleringsmodell med $K = 128$ faktorer.



Figur 11. Sekventiell bifurkation finner de $k = 3$ viktigaste faktorerna x_{68}, x_{113} och x_{120} bland $K = 128$ parametrar efter 16 simuleringar $y_{(\cdot)}$ [9].

Vi noterar att för att finna de k viktigaste faktorerna i en simuleringsmodell med K faktorer krävs som mest

$$1 + k \log_2(2K / k)$$

simuleringar [9]. Vi kan som exempel finna de åtta viktigaste faktorer i en modell med 512 faktorer efter 57 simuleringar.

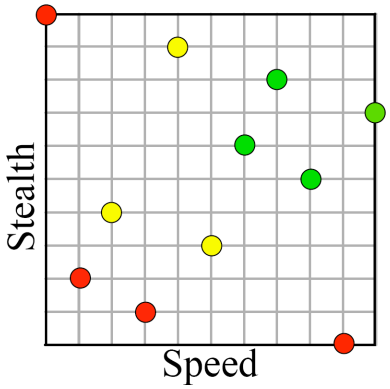
Ofta svarar ett litet antal faktorer för det mesta av variansen i simuleringsmodellens responsfunktion. Vi kan då fokusera efterföljande simuleringar på dessa faktorer och behandla övriga faktorer som konstanter.

2.1.5 Nästan ortogonal latinska hyperkuber (NOLH)

2.1.5.1 Problembeskrivning

I explorativ dataanalys önskar vi designs som kan passa en mängd olika simuleringsmodeller. För sådana situationer har Latinska hyperkuber (LH) visat sig vara en ovärderlig teknik. Detta är den dominerande designen för experiment med datorsimuleringar. En viktig anledning till detta är att de kommer med minimala begränsningar avseende antalet faktorer. Denna flexibilitet omfattar även analys och visualisering av simulatorutdata.

I figur 12 visas ett exempel på en LH för problemet med hastighet och smygförmåga. Vi ser att samma budskap som i figur 7 i huvudsak förmedlas genom figur 12 med väsentligt färre indatapunkter.



Figur 12. En LH för exemplet med hastighet och smygförmåga [6].

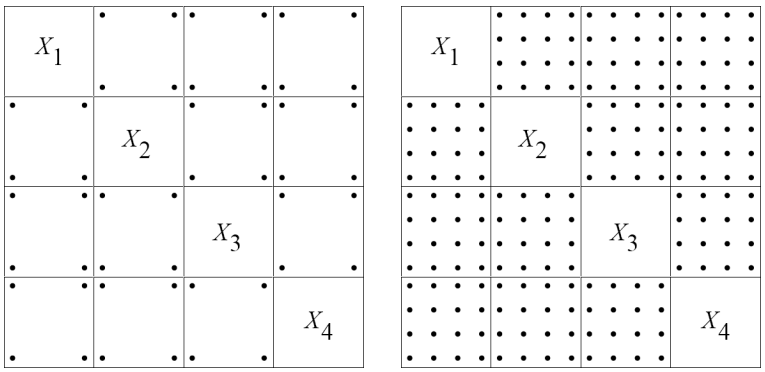
För att förhindra oacceptabel korrelation mellan modellens faktorer vilket hindrar dataanalysen för många statistiska metoder, t.ex. regressionsanalys och regressionsträd, har algoritmer utvecklas som konstruerar nästan ortogonala Latinska hyperkuber (NOLH) som minskar eller eliminerar korrelationer mellan ingående faktorer.

I figur 2 (övre högra hörnet) ser vi att NOLH passar bäst till stora simuleringsmodeller med många faktorer och olinjär responsfunktion. Speciellt är detta en lämplig metod när man inte kan göra några antaganden om faktorerna eventuella interaktioner och inte vet något precist om eventuella olinjäriteter.

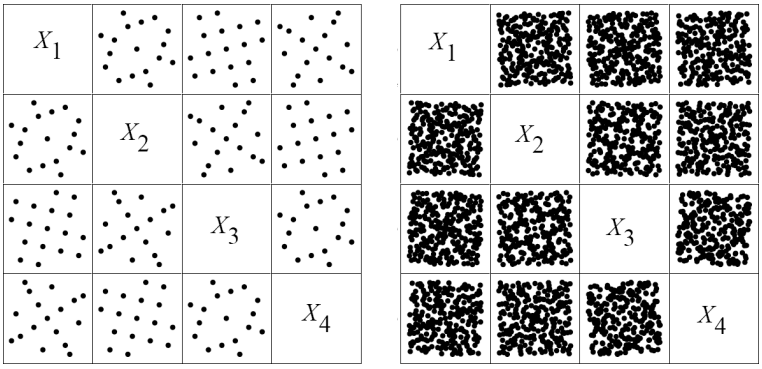
2.1.5.2 Metodbeskrivning

NOLH-designs är viktiga på ett antal olika sätt, de är mycket effektiva i antalet körningar n som krävs för att undersöka antalet faktorer k som är av intresse. Eftersom dessa designs vanligen har god rymdfyllning och ortogonalitet, är de ett attraktivt alternativ för att utforska en okänd responsfunktion. Antalet indatapunkter är radikalt färre än för faktordesigns. Till exempel kan 20 faktorer undersökas i en NOLH med endast 129 indatapunkter.

Spridningsdiagram för fyra olika designs visas i figurerna 13 och 14. Dessa är en 2^4- och en 4^4-faktordesign, en NOLH design med 17 indatapunkter, och en NOLH design med 257 indatapunkter. I figurerna visas rymdfyllanden i tvådimensionella projektioner för de fyra designmetoderna. Rymdfyllnaden hos en NOLH med 17 indatapunkter står sig väl i jämförelse med den för 4^4-faktordesignen med $^1/_{15}$ av beräkningstiden. Alternativt kan analytikern använda en NOLH med 257 indatapunkter och få möjlighet att undersöka en mycket tätare uppsättning kombinationer av faktorernas nivåer i jämförelse med de 256 indatapunkter i en 4^4-faktordesign. Fördelarna med att sampla indata med en NOLH är som störst vid stora k. Att göra flera körningar på samma design låter oss avgöra variansen i simuleringens responsfunktion.



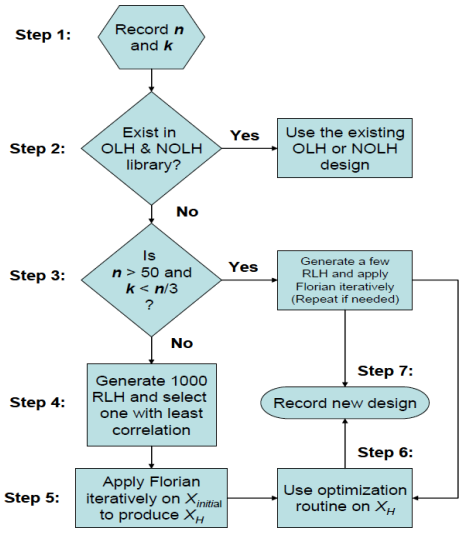
Figur 13. Två-dimensionella projektioner av 2^4- och 4^4-faktordesigns [14].



Figur 14. Två-dimensionella projektioner av NOLH med 17 respektive 257 indatapunkter för analys av fyra faktorer [14].

Vi kan använda heltalsprogrammering för att konstruera NOLH vilket undanröjer tidigare dimensionalitetsproblem vid konstruktionen av NOLH [10]. Det är möjligt att konstruera NOLH för hundratals faktorer och körningar [11][12][13].

Metodiken för konstruktion av NOLH bygger på en kombination av att generera NOLH med heltalsprogrammering där NOLH utvärderas utifrån sin ortogonalitet och rymdfyllande förmåga, figur 15.



Figur 15. Konstruktion av NOLH för kontinuerliga faktorer [11].

Ortogonalitet mäts med $\rho_{map} = \max\left\{\left|\rho_{ij}\right|, \forall(i \neq j)\right\}$ där ρ_{ij} är ortogonaliteten mellan indatavektorerna för två faktorer x_i och x_j. Rymdfyllnad mäts med ett mått kallat ”modifierad L_2 diskrepans” (ML_2). För detaljer om optimeringen med heltalsprogrammering och utvärderingen av konstruerade NOLH med ρ_{map} och ML_2 hänvisas till [11]. Metodiken konstruerar godtyckliga NOLH där $n < k$ för simuleringsmodeller där *alla faktorer är kontinuerliga* och har samma antal nivåer för varje faktor.

2.1.6 NOLH för diskreta och kategoriska faktorer

Om simuleringsmodellen innehåller en blandning av *kontinuerliga och diskreta faktorer* som dessutom kan ha olika antal nivåer per faktor kan man använda en metod som utvidgar metodiken för att konstruera NOLH till att även hantera diskreta faktorer [15]. Innehåller modellen även *kategoriska faktorer* kan man använda en metodik som kallas nästan ortogonal, nästan balanserade mixad design (NONB) [16]. Båda dessa metoder utgår ifrån och generaliserar [11][12].

2.1.7 Översikt över designmetoder

En tabell med en översikt över alternativa designmetoder karakteriserade utifrån antalet faktorer i simuleringsmodellen, vilken typ av faktor som ingår i modellen (kontinuerliga, diskreta, kategoriska), samt vilken typ av responsfunktion som ges av simuleringsmodellen presenteras av Sanchez och Wan [14], se figur 16.

Denna tabell kan utgöra ett första steg i att mappa en aktuell simuleringsmodell mot en effektiv experimentdesign. Har simuleringsmodellen en okänd kanske olinjär responsfunktion med ett stort antal faktorer, troligen med komplicerade interaktionstermer rekommenderas en NOLH (eller NONB).

	2^k factorials	m^k factorials, $3 <= m <= 5$	m^k factorials, $6 <= m <= 10$	R3FF, orthogonal arrays	Foldover designs	R4FF, Plackett-Burman designs	R5FF	Central composite with full factorial	Central composite with R5FF	random LH with $n >> k$	smallest possible NOLH (i.e., very few extra columns)	larger NOLH	crossed NOLHs	FFCSB (main effects)	FFCSB-X or Hybrid method
FACTOR CHARACTERISTICS															
Total number of factors: 2-6	B* ●	L* ●	●	●	●	●	●	●	●	●	●	C* ■	○3	○	○
Total number of factors: 7-10	●	○1		●	●	●	B* ●	●	●	●	●	C* ■	○3	○	○
Total number of factors: 11-29				●	●	●	B* ●		●	●	●	C* ■	○3	○	○
Total number of factors: 30-99				●	B* ●	●	○2		○2	●4			●	●1	●
Total number of factors: 100-300				●	B* ●	●	○2		○2	●4			○3	●1	●
Total number of factors: 300-1000				●	○									●1	B*C* ●
Total number of factors: 1000-2000				●	○									●1	B*C* ●
Binary factors	B* ●			●	●	●	●							●	
Qualitative factors with 3 or more levels		L* ●	●												
Discrete or continuous factors treated as binary	○1			○1	○1	○1	○1							●	
Discrete factors, 3-5 levels of interest	○1	●						●	L* ●						
Continuous factors, or discrete with many levels		●	●					●	●	●	●	●	●		●2
Decision factors (controllable in real world)	●	●					●				●	●			
Noise factors (uncontrollable in real world)				●	○4	○4	○4			○5	●				
RESPONSE CHARACTERISTICS															
Main effects only (initial screening)	○2			●1	■	■	■		■		●	○6		●1	■
Main effects (valid w/ 2-way interactions exist)	○2				●			□2	■						●
Main effects and all 2-way interactions	○2	■	■				●	■	●				●		
Main effects and many interactions	●	○2	○2					●		●5	●5	●5	●		
Quadratic effects		○2	○2					●	●	●	●	●	●		
Thresholds / non-smooth effects		○2	○2							●	●	●	●		
Flexible modeling - not all pre-specified		○2	○2							●	●	●	●		
OTHER CONSIDERATIONS															
Batch mode unavailable - all runs through GUI		○2		●3					●3		●3				

■ Provides additional modeling flexibility or allows some assumptions to be assessed

B* Good design choice for binary factors

L* Good design choice for factors with a limited number of qualitative or discrete levels

C* Good design choice for continuous factors, discrete factors with many levels

● Works well

- ●1 Assumes that interactions are negligible or that they'll show up with the main effects - must follow up with confirmation runs
- ●2 For FFCSB-X, "many" means 2 or 3 levels
- ●3 Smaller designs are the only ones feasible until this gets "fixed" - work with the developer
- ●4 Design's correlation structure must be checked - stacking many designs may be an alternative
- ●5 Degrees of freedom limit the number of terms that can be estimated simultaneously, so not all main effects and two-way interactions can be estimated simultaneously.

○ Consider these designs if additional computing resources are available

- ○1 These require many more runs than other designs unless k is small. Consider NOLH designs.
- ○2 Start with 2 replications and see if you can eliminate any factors - each time you do, you effectively double the number of replications for factors that remain.
- ○3 Same as above, but to avoid overly-large designs you may want to consider saturated or nearly-saturated NOLH

□ Potential designs that provide additional modeling flexibility or allow some assumptions to be assessed, but typically require many more design points

○ Potential designs, but better designs exist for this purpose

- ○1 Unless used for initial screening, it may be a good idea to explore at least 3 levels
- ○2 These require many more runs than other designs unless k is small. Consider R5FF (for binary) or NOLH designs
- ○3 Easier to use a larger NOLH (if all factors are quantitative) or else cross a full factorial for factors with just a few levels with an NOLH
- ○4 Since you do not need to estimate interactions among noise factors, use a screening design like R3FF or a small NOLH
- ○5 In the spirit of keeping noise factor designs small, you might prefer an NOLH
- ○6 If you're interested in screening and want to keep the number of runs down, go for one of the smaller LH designs

Figur 16. Alternativa experimentdesigns [14].

NOLH-designs för upp till 29 faktorer finns färdiga för nedladdning i ett excelark från NPS SEED Center[2].

[2] SEED Center software (2011). [Online] http://harvest.nps.edu/software.html

3 Dataanalys och visualisering

När data samlats in, experimentdesign genomförts och simuleringarna slutligen körts har ofta en väldigt stor mängd data genererats – ett resultat som kan vara såväl svårhanterligt som svåröverskådligt. Vi är intresserade av att utröna vad det egentligen är som resultaten innebär och vilka slutsatser som kan dras från vårt underlag. Detta kan i sin tur generera en mängd olika frågor som behöver besvaras och exempel på områden som vi kan vilja kasta ljus över är:

- systemsvarens (utdatas) spridning,
- centraltendenser hos systemsvaren,
- förhållanden mellan systemsvar,
- hur olika faktorer (indatavariabler) påverkar varandra och systemsvar,
- intressanta områden eller tröskelvärden för faktorerna,
- generella kännetecken hos landskapet av möjligheter.

Hur kan då information gällande ovanstående punkter vara oss behjälpligt? Genom att endast modellera och simulera vet vi förvisso hur ”byggstenarna” ser ut och vilka resultat som genereras – däremot har vi ofta väldigt liten kunskap gällande det inre system, den dynamik, som faktiskt orsakar de utfall som observeras. Genom tillämpning av användbara metoder inom analys och visualisering av utdata kan problemförståelsen fördjupas och svar finnas.

Inom data farming används ofta en blandning av tekniker från områdena visualisering, data mining och statistisk analys. Statistiska analystekniker används för att utforska de tidigare nämnda punkterna – exempelvis genom att karaktärisera fördelningar, sätta upp konfidensintervall och genomföra hypotestester. Data mining används istället för att söka efter mönster och förhållanden och visualiseringstekniker kan utgöra kraftfulla verktyg för att undersöka, utforska och presentera data. Att undersöka data genom visualisering kan exempelvis fungera som en valideringsmetod men också för kvalitetssäkring av data och använda visualisering för utforskning bidrar istället till att finna ny information och nya insikter. Slutligen handlar presentation mer om att förmedla budskap och förståelsen för hur data och resultat skall presenteras så att budskapet blir lättförståeligt och lättillgängligt. Teknikerna och deras syften kan med andra ord se ganska olika ut – gemensamt för dem alla är dock utgångspunkten att en bra simulering skall vara effektiv, precis, estetisk och anpassningsbar.

Istället för förlita sig till någon eller några enstaka tekniker är inställningen inom data farming att använda en kombination av olika metoder där varje tekniks speciella styrkor utnyttjas på ett sådant effektivt sätt som möjligt. Helt åtskilda är inte de olika områdena utan de överlappar varandra – både sett till ingående tekniker och till den information man eftersträvar att utvinna. Ofta kan det exempelvis vara en god idé att inleda en analys av ett problem som inkluderar ett stort antal inverkande faktorer med visualiseringar för att nå en mer övergripande helhetsförståelse. Detta kan sedan följas av tillämpning av mer statistiska analysmetoder för att sedan inkludera ytterligare visualisering – helt enkelt en växelverkan där syftet styr metodvalet.

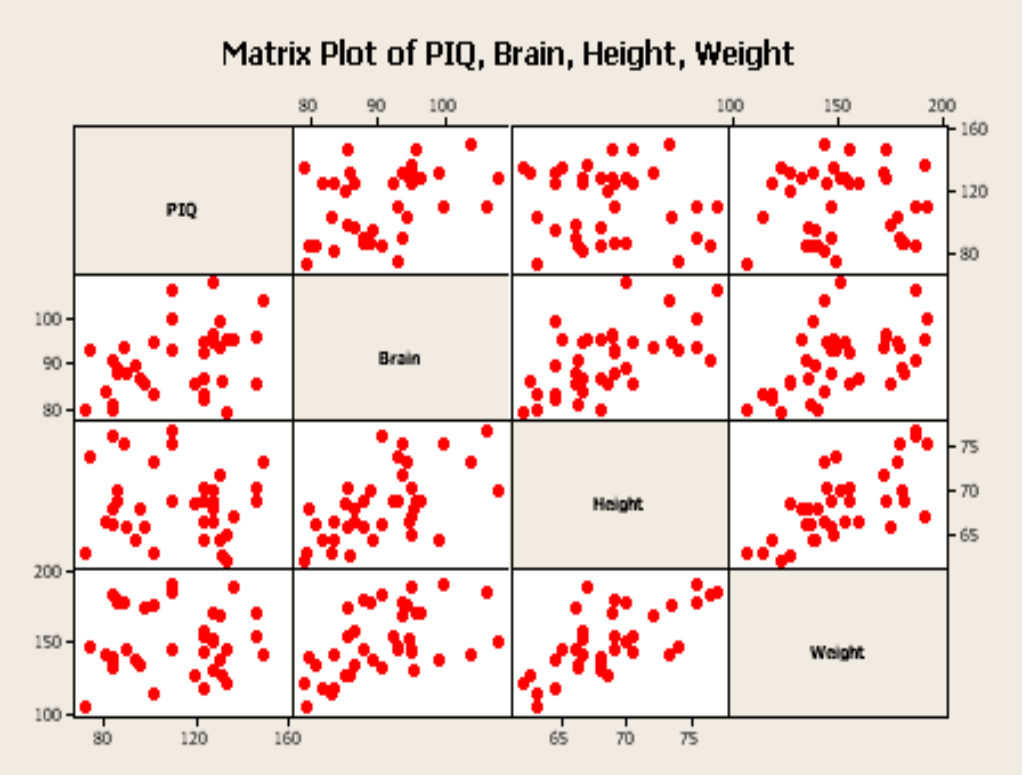
I detta kapitel följer en generell beskrivning av ett antal metoder och tekniker som ofta används för analys och visualisering av utdata inom data farming. Materialet som presenteras inleds med en del mer analytiska tekniker såsom regressionsanalys i olika former och variansanalys. Inriktningen övergår sedan alltmer mot mer renodlat multivariata tekniker, såväl analytiska som för visualisering, exempelvis principalkomponentsanalys och glyfer. Den blandning av metoder som presenteras täcker ett ganska brett spektra, både gällande tillämpningsområden och svårighetsgrad. De utgör tillsammans en användbar grund för fortsatt framtida utforskning inom området för data farming.

3.1 Metoder

3.1.1 Spridningsdiagram

3.1.1.1 Problembeskrivning

Spridningsdiagramstekniken används för att plotta tvådimensionella data så att den horisontella axeln visar värdena hos den ena variabeln och den vertikala axeln visar den andra variabelns värden. Genom detta kan förhållanden mellan variablerna studeras och analyseras. Tekniken kallad matris av spridningsdiagram (eng. scatterplot matrix) och används istället för att studera alla möjliga variabelpar i en flerdimensionell datamängd. Förutom att upptäcka förhållanden mellan värden kan tekniken också användas för att upptäcka mönster som sträcker sig över förhållandena [17][18][19]. Matriser av spridningsdiagram kan även användas för att detektera anomalier [20]. Spridningsdiagramsmatrisen är helt enkelt en serie av spridningsdiagram som åskådliggörs simultant för att ge en så omfattande och heltäckande bild av systemet som möjligt och som vi ser i figur 17 tillhandahåller diagrammet en överblick som användaren snabbt kan tillgodogöra sig.

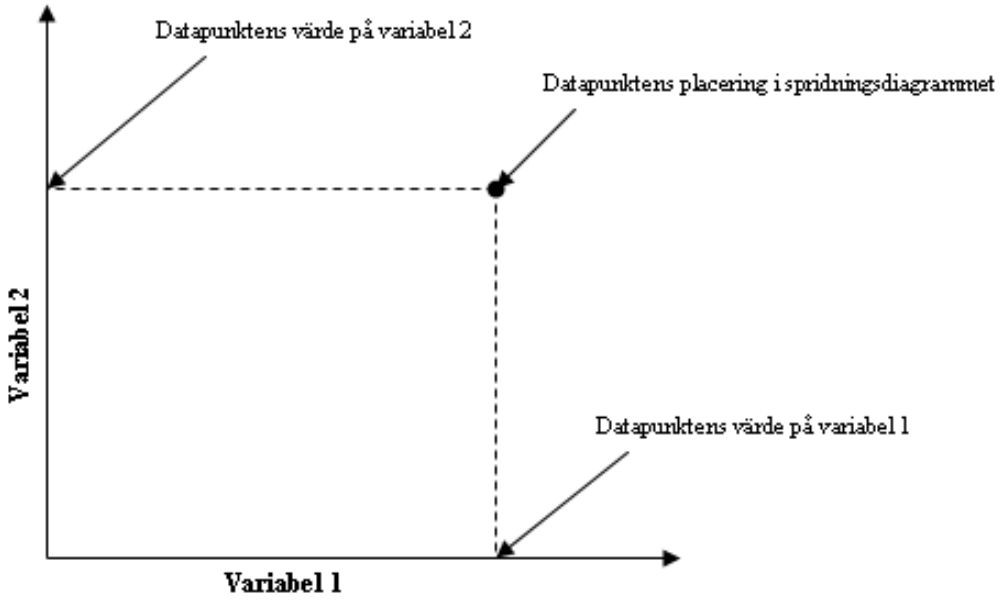


Figur 17. Matris över spridningsdiagram [21].

Om modellen innehåller k stycken variabler så finns det $k(k-1)/2$ par vilket innebär att antalet spridningsdiagram kan bli stort även för ett relativt litet antal variabler. Matriser av spridningsdiagram är sammanfattningsvis användbara för att snabbt utröna bivariata förhållanden, men denna överblick kan försvåras då många variabelpar innebär många små individuella spridningsdiagram vilket helt enkelt kan göra det svårt att se eventuella förhållanden [19].

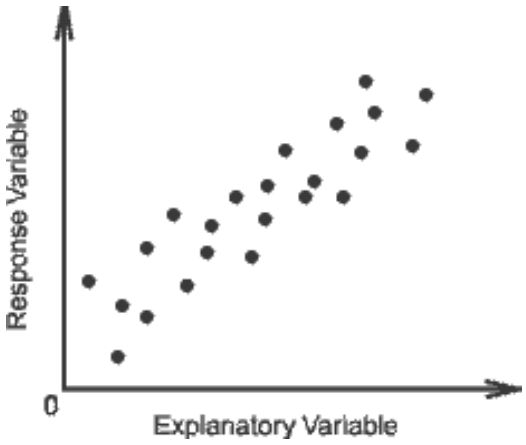
3.1.1.2 Metodbeskrivning

Vi bygger vårt spridningsdiagram genom att rita ut en datapunkt så att den motsvarar rätt värde på båda variabelaxlarna vilket visas i figur 18.



Figur 18. Ett spridningsdiagram innehållande en tvådimensionell datapunkt [22].

Genom ett spridningsdiagram som inkluderar alla datapunkter för variabelparet i datamängden kan vi exempelvis åskådliggöra förhållanden såsom korrelation variablerna emellan. Figur 19 exemplifierar ett förhållande som är positivt korrelerat.



Figur 19. Spridningsdiagram av ett tvådimensionellt förhållande med positiv korrelation [23].

Några hjälpmedel som kan vara bra att använda vid individuella spridningsdiagram är exempelvis möjligheten att lägga till symboler och inkludera konfidensintervall för regressionslinjer – dessa hjälpmedel bidrar dock ofta till ett alltför rörigt intryck om de inkluderas i en matris av spridningsdiagram [19].

3.1.2 Regressionsanalys

3.1.2.1 Problembeskrivning

För att modellera förhållandet mellan en responsvariabel och en eller flera förklarande variabler kan regressionsanalys användas, och därigenom utforskas och kvantifieras de eventuella samband som kan existera mellan de olika variablerna. Ofta finns ett intresse av att förstå den kausalitet som råder mellan variablerna och målet är då att försöka förstå hur värdet på responsvariabeln förändras då de förklarande variablerna varieras [24][25].

3.1.2.2 Metodbeskrivning

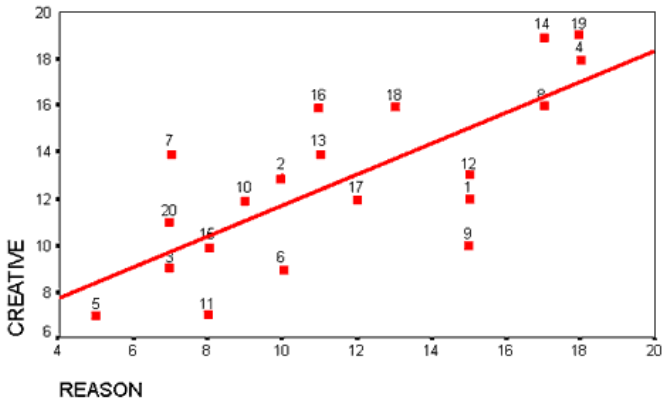
Vi inleder genomförandet av en regressionsanalys genom att en lämplig modell för det aktuella problemet väljs ut, data om variablerna samlas in, och modellen anpassas sedan så att den återspeglar verkligheten på ett så bra sätt som möjligt, vilket ofta sker genom tillämpning av minstakvadratmetoden [26]. Regressionen används för att uppskatta huruvida, och i så fall på vilket sätt, den eller de förklarande variablerna påverkar responsvariabeln. Resultatet blir följaktligen ett uppskattat förhållande vars statistiska signifikans sedan bestäms [24]. Förhållandet mellan variablerna utforskas sedan ytterligare och används även för att uppskatta eller prediktera det förväntade systemsvaret för ett givet värde hos den eller de förklarande variablerna [26].

Regressionsanalyser brukar inledas med att en arbetshypotes gällande förhållandet mellan de aktuella variablerna formuleras. Det är denna hypotes som man sedan utforskar under analysen. Genom att rita upp ett spridningsdiagram (se föregående avsnitt) eller en matris av spridningsdiagram kan systeminformation i form av exempelvis olika samband eller anomalier komma att urskiljas. Informationen kan förhoppningsvis underlätta valet av modell och regressionsanalysen fungerar sedan som ett undersökande statistiskt verktyg där den metod som tillämpas beror på vilken regressionsmodell som valts [24]. En hel del olika metoder återfinns således under paraplyet för regressionsanalys, varav enkel, multipel och stegvis regression utgör några ofta förekommande exempel inom data farming.

Enkel regression

Enkel regression är ett specialfall av multipel regression och används när endast en förklarande variabel är inkluderad. Ofta görs ett antagande om ett linjärt samband vilket leder till att regressionens uppgift reduceras till att uppskatta de icke-observerbara parametrarna. Då brustermen antas vara noll i genomsnitt blir den resulterande uppgiften helt enkelt att uppskatta var den räta linjen associerad med modellen är lokaliserad [24].

Genom att använda sig av exempelvis minstakvadratmetoden kan den så kallade regressionslinjen (den linje som bäst beskriver sambandet mellan variablerna då den minimerar summan av de kvadrerade differenser som återfinns mellan linje och observationer) ritas upp, vilket exemplifieras i figur 20.



Figur 20. Regressionslinje framtagen med minstakvadratmetoden [27].

Mer specifikt genereras skattningar för interceptet (skärningspunkten med y-axeln) och riktningskoefficienten i linjens ekvation [24]. För att minstakvadratmetoden skall kunna användas krävs dock att en del antaganden görs:

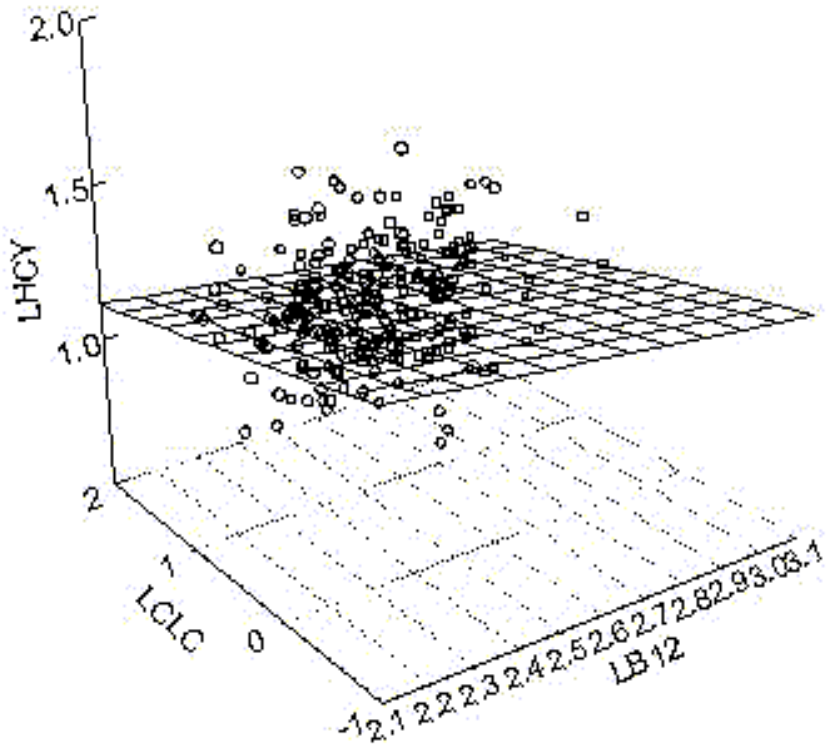
- alla brustermer har väntevärde noll och konstant varians,
- alla brustermer är normalfördelade,
- alla brustermer är oberoende slumpvariabler (endast av betydelse då tidsseriedata behandlas).

Dessa antaganden kontrolleras sedan innan analysen fortsätter, och givet att de är uppfyllda krävs även en utvärdering av hur väl modellen anpassar sig till data innan modellen tas i bruk. De linjära sambanden kan undersökas ytterligare genom t ex kovarians och korrelationskoefficient och eventuella anomalier undersöks också för att försöka förstå deras uppkomst och betydelse. Ofta är man intresserad av att testa huruvida de uppskattade parametrarna är signifikant skilda från de sanna vilket innebär genomförandet av *t*-tester (hypotestester som genomförs för att avgöra om två populationer skiljer sig åt). Förklaringsgraden, R^2, används som ett mått för att utvärdera modellen och kommer att förklaras senare i avsnittet [24][28].

Godkänns modellen för användning blir nästa steg exempelvis att konstruera konfidensintervall för den sanna riktningskoefficienten respektive interceptet, vilket görs med hjälp av medelfelen för skattningarna, eller att ta fram en prognosmodell [28][29]. För att göra detta extrapoleras helt enkelt den framtagna regressionslinjen.

Multipel regression

Vid multipel regression innehåller modellen två eller fler förklarande faktorer och är egentligen ett specialfall av *the General Linear Model* [30]. Den grundläggande logiken är densamma som för enkel linjär regression med skillnaden att det nu handlar om att uppskatta ett regressionsplan (i fallet för två förklarande variabler), vilket visas i figur 21, istället för en regressionslinje [28].



Figur 21. Regression med synliggjort regressionsplan för två förklarande variabler [31].

Multipla regressionsmodeller anpassas precis som enkla regressionsmodeller genom t ex. minstakvadratmetoden och som är fallet vid enkel regression skall samma antaganden göras även här [24][32][33][34]:

- alla brustermer har väntevärde noll och konstant varians,
- alla brustermer är normalfördelade,
- alla brustermer är oberoende slumpvariabler.

Om antagandena efter kontroll visar sig vara uppfyllda, och eventuella anomalier undersökts, blir nästa steg även här att undersöka hur väl modellen passar till data och på så sätt utvärdera modellen. Genom att uppskatta brustermens varians kan förklaringsgraden hos regressionsanalysen och den statistiska signifikansen hos dess parameteruppskattningar utvärderas med hjälp av *F*-test (se avsnitt om variansanalys för utförligare beskrivning) [24].

Om modellutvärderingen tyder på en godkänd modell så kan skattningarna tolkas för att på så sätt få djupare information gällande relationerna mellan respons- och förklarande variabler. Modellen kan också användas för att göra prediktioner och uppskattningar genom beräkningar av prediktions- respektive konfidensintervall [35].

Stegvis regression

Stegvis regression används när man har ingen eller liten förkunskap om systemet och är en så kallad modellgenererande teknik. Den kan ses som en systematisk metod för att addera och ta bort faktorer från en multilinjär modell baserad på deras statistiska signifikans i en regression [36][37]. Metoden utgår från en ”grundmodell” och jämför sedan förklaringsgraden hos ökande och minskande modeller.

Stegvis regression kan åstadkommas antingen genom att prova ut en oberoende variabel åt gången och inkludera den i regressionsmodellen, eller genom att inkludera alla potentiella oberoende variabler i modellen och sedan avlägsna de som inte är statistiskt signifikanta. En kombination av båda varianterna kan också tillämpas [38].

Det vanligaste kriteriet för att addera eller ta bort en variabel är baserat på den så kallade partiella *F*-statistikan (eng. partial *F*-statistic) [32]. För varje steg beräknas *p*-värdet för *F*-statistikan för att testa modellen. Förenklat sett kan vi säga att beroende på om vi har valt att addera eller ta bort variabler så väljs den variabel vars *F*-statistika medför det minsta respektive största *p*-värdet. Om hypotestest visar att termens koefficient är eller blir noll (beroende på om variabeln redan ingår i modellen eller ej) så skall variabeln inte inkluderas [37].

3.1.2.3 Begrepp

Goodness of fit

Ett annat sätt att avgöra huruvida en regressionsmodell är användbar för att förutsäga värdena på responsvariabeln är genom modellens ”goodness of fit”, vilket kort och gott säger oss hur väl vår modell anpassar till data. Goodness of fit kan exempelvis utvärderas genom att använda ett *F*-test i form av variansanalys, vilket beskrivs i senare avsnitt [32].

R^2

R^2, förklaringsgraden, är ett mått på den andel av den totala variationen hos responsvariabeln som förklaras av regressionen – ett mått på Goodness of Fit. Ett högt R^2 indikerar att regressionsmodellen förklarar variationen hos responsvariabeln väldigt bra, medan ett lågt R^2 däremot indikerar att det kan finns viktiga faktorer som utelämnats från regressionsmodellen [24][32]. Justeras antalet frihetsgrader får man ett så kallat justerat R^2, vilket ofta är ett mer användbart mått då det möjliggör jämförelse av förklaringsgrad mellan modeller som har skilda antal förklarande variabler [32][39]. Genom att lägga till förklarande variabler i sin modell kan R^2-värdet förbättras men då riskerar man också att modellens enkelhet går förlorad likväl som generaliserbarheten till andra datamängder. Dessa ”fallgropar” kommer sig av s.k. överanpassning (eng. overfitting) av data som uppkommer då modellen börjar beskriva brustermerna istället för den faktiska relationen, vilket bland annat kraftigt kan reducera modellens prediktiva förmåga [40].

Parametriska och icke-parametriska metoder

Linjär regression utgör ett exempel på en parametrisk regressionsmetod – en metod där man gör antaganden om att data kommer från någon sannolikhetsfördelning samt gällande de ingående parametrarnas medelvärde, standardavvikelse etc. Den parametriska

regressionsmetodens fördel ligger främst i dess relativa enkelhet men detta medför också metodens nackdel, nämligen de relativt strikta antaganden som krävs för att hålla metoden enkel. En icke-parametrisk (även kallad fördelningsfri) metod kan användas när dessa antaganden inte går att uppfylla. Om spridningsdiagrammet inte erbjuder någon ökad insikt i variabelsambanden kan ett bra hjälpverktyg vara LOESS (en icke-parametrisk regressionsmetod) eftersom metoden jämnar ut data till en jämn kurva i spridningsdiagrammet [41]. Mer information om icke-parametrisk regression och LOESS följer i senare avsnitt.

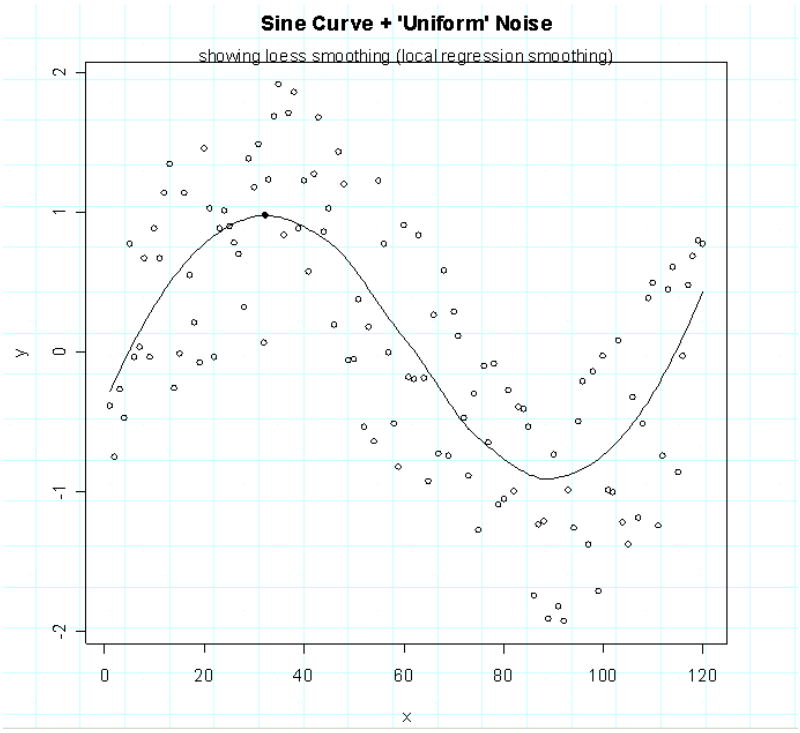
3.1.3 Icke-parametrisk regression - LOESS

3.1.3.1 Problembeskrivning

Som tidigare nämnt kan icke-parametrisk regression utgöra ett användbart alternativ till parametriska regressionsmetoder när det inte är möjligt att göra de antaganden som är nödvändiga vid t.ex. linjär regression. De icke-parametriska regressionsmetoderna kräver helt enkelt inte lika stränga antaganden gällande regressionsfunktionen som de parametriska. Nackdelen med de mindre stränga villkoren är större beräkningskostnader och ibland även ett mer svårtolkat resultat. Fördelen är istället en potentiellt mer noggrann skattning av regressionsfunktionen [42].

Det finns flera icke-parametriska regressionsmetoder, varav LOESS (eng. locally weighted scatterplot smoothing) är en av dessa [41]. LOESS är en teknik som används för att anpassa en regressionsyta till data genom så kallad flervariat utjämning. Responsvariabeln jämnas då ut som en funktion av den förklarande på ett ”glidande vis” likt hur glidande medelvärde beräknas för en tidsserie. Utjämningsprocessen kallas ”lokal” eftersom varje utjämnat värde bestäms av de närliggande datapunkterna inom ett visst intervall. LOESS kan också användas för att fylla i saknade data [43][44].

Figur 22 visar hur den utjämnade medelvärdeslinjen för en specifik datamängd kan se ut. Det kan vara svårt att se eventuella trender och mönster då vi hanterar stora mängder eller väldigt brusig data men genom att använda LOESS kan vi uppskatta en trendkurva för att öka vår förståelse för datamängden [45].



Figur 22. Data och LOESS-utjämning [44].

3.1.3.2 Metodbeskrivning

Linjäritetsantagandet vid icke-parametrisk regression är alltså inte fullt så strikt som för parametrisk utan istället räcker antagandet att *f(x)* är en jämn funktion; *f(x)=f(-x)*. Funktionen *f(x)* specificeras inte heller på förhand vilket är fallet vid linjär regression. Dock är antaganden gällande slumptermen oftast desamma för icke-parametrisk som för linjär regression, se metodbeskrivning för Regressionsanalys [41].

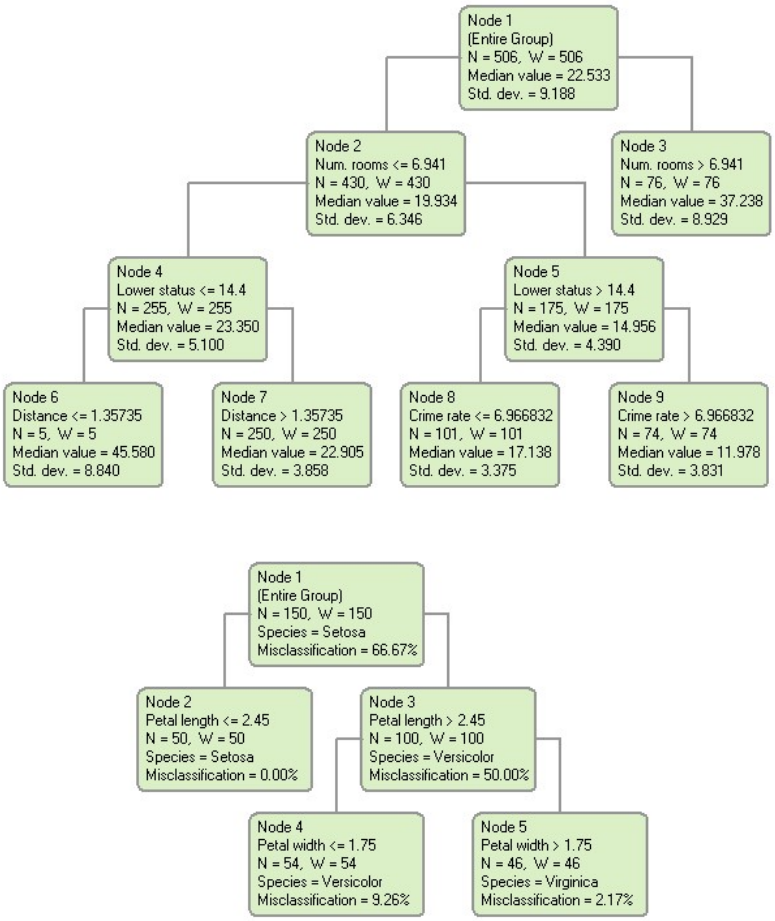
En enkel metod för att förbättra skattningen av $f(x_i)$ är Binning, vilket är en metod som delar in x i många och små intervall. En alternativ metod är lokalt medelvärde (eng. Local Averaging) som till stor del liknar Binning, med skillnaden att metoden använder sig av ett intervall som flyttas kontinuerligt över data istället för icke-överlappande intervall. Grundtanken bakom dessa, och också följande, metoder är att observationer nära något värde x är informativa om *f(x)*. Ibland är dock skattningen av *f(x)* udda trots ett antagande om att den sanna regressionsfunktionen är jämn. För att komma till rätta med det problemet kan kärnregression (eng. Kernel regression) användas, där problemet löses genom att x_i viktas. LOESS kan sägas vara en utveckling av kärnregression och anpassar ett polynom (av låg grad) i x genom att använda de vikter som tilldelas genom kärnregressionen. Metoden kan användas vid såväl en som flera förklaringsvariabler. Genom den viktade minstakvadratmetoden (eng. Weighted Least Squares, WLS) anpassas ekvationen så att den viktade residualkvadratsumman minimeras och anpassningen förbättras. Proceduren upprepas sedan för alla x eller för ett antal representativa x [41].

Den proportion observationer som används för varje skattning av *f(x)* kallas utjämningsparametern. Parametern betecknas med α och väljs oftast genom att skattningen på *f(x)* utvärderas grafiskt. Vårt mål är en jämn funktion som skattas genom ett så litet α som möjligt. Om skattningen inte är jämn nog så återupprepas processen med ökande α till dess att en önskad nivå av jämnhet uppnåtts. Är funktionen däremot jämn redan initialt så minskas α så länge jämnheten bibehålls [41].

3.1.4 Klassificerings- och regressionsträd (CART)

3.1.4.1 Problembeskrivning

CART (Classification and Regression Tree) är en statistisk klassificeringsmetod baserad på rekursiv partition som konstruerar så kallade beslutsträd vilka bl. a. kan hjälpa oss till större insikter gällande faktorers effekter på utdata samt för att klassificera nya data [46]. Såväl klassificering som regression involverar prediktion av en responsvariabel givet ett antal förklarande variabler. Är responsvariabeln kontinuerlig eller diskret och kan anta reella värden så är problemet av regressionstyp medan en kategorisk responsvariabel resulterar i ett klassificeringsproblem [47]. I figur 23 exemplifieras visualiseringar av de olika sorters träden.



Figur 23. CART-träd – det övre trädet visar ett regressionsträd medan det undre visar ett klassificeringsträd [48].

Ett beslutsträd representeras av en uppsättning av binära frågor som förgrenar urvalet i allt mindre delar och trädet indikerar då vilka faktorer som är viktigast i modellen. Informationen som kan utläsas kan också användas för att t.ex. avgöra huruvida det finns någon skillnad i de signifikanta faktorerna som identifierats med en passad modell jämfört med samtliga rådatapunkter [49].

Metodens användbarhet ökar då den kan hantera såväl numeriska som kategoriska variabler. En annan fördel är även dess robusthet mot anomalier – ofta brukar nämligen fördelningsalgoritmen isolera anomalier i en individuell nod eller noder. Ytterligare en viktig egenskap hos CART är att strukturen hos dess träd är invariant med avseende på monotona transformationer[3] av förklarande variabler. I och med detta kan vilken variabel som helst bytas ut mot dess logaritm eller kvadratrotsvärde utan att trädets struktur förändras [46].

Utöver de fördelar som redan nämnts finns även en hel del andra, varav den kanske främsta styrkan av dem alla är att visualiseringarna av träden är relativt lätta och intuitiva att tolka – något som gör det möjligt för en bredare grupp av användare att förstå den bakomliggande modellen [49][51]. Dock kan stora träd snabbt bli klumpiga och svårhanterliga och riskerar då att tappa det inneboende förklaringsvärdet. Användaren bör också vara medveten om att trädstrukturer kan vara instabila på det sättet att en förändring i urvalet kan resultera i väldigt olika träd [52].

[3] En monoton transformation innebär en transformation av en strikt växande funktion [50].

3.1.4.2 Metodbeskrivning

En CART-analys består ofta av tre delar [46]:

1. framtagande av det maximala trädet,
2. val av rätt trädstorlek,
3. klassificering av ny data genom att använda det konstruerade trädet (beroende på syftet med analysen kan detta steg ibland utgå).

Inom Data Farming används ofta de två de första två delarna, vilka här beskrivs mer detaljerat.

Framtagande av det maximala trädet

Innan ett träd skapas bestäms med vilket kriterium de delningar som ska utföras ska väljas ut och även stoppvillkoret som ska gälla för delningen bestäms.

Inför en förgrening genomsöks alla möjliga variabler och värden för att hitta den bästa delningen, vilket är den som delar in data i två undergrupper med maximal homogenitet, och på så sätt förbättra passningen. Denna process upprepas sedan för varje resulterande förgrening. Maximal homogenitet definieras genom den s.k. ”impurity-funktionen” (eng. impurity function eller impurity criteria). Det s.k. Gini Index är ett av de mest använda måtten på ”impurity” inom metoden – andra mått är t ex ”the Twoing splitting rule” och entropi [46][47][51]. För passning av regressionsträd används dock inte Gini Index utan istället summan av kvadraterna, och på så sätt väljs de delningar som orsakar den största minskningen i summan av kvadraterna ut [46].

Vilket stoppvillkor som används varierar men kan t ex vara att det endast finns en observation i de noder som ska förgrenas, att alla observationer i noden har en identisk fördelning av förklarande variabler vilket omöjliggör en förgrening, eller helt enkelt att en gräns som satts för antalet nivåer i trädet har nåtts [46].

Val av rätt trädstorlek

Kompromissen mellan systematiskt fel (eng. bias) och varians för modeller handlar om hur stort trädet skall göras. Ett stort träd kommer att förgrena data till mindre och mindre delar vilket förmodligen kommer att resultera i överanpassning (eng. overfitting), men å andra sidan så kanske ett litet träd inte är utvecklat nog att fånga de viktiga relationer som kan finnas variablerna emellan [53]. Det maximala träd som skapats överanpassar alltså sannolikt data, trots detta är det vanligaste tillvägagångssättet ändå att man inledningsvis tar fram ett maximalt träd och istället sedan beskär (eng. prune) det till en storlek som skapar en bra balans mellan anpassning och överanpassning [46][54].

Genom trädbeskärning skapas en serie av träd med allt enklare struktur genom att noder av ökande inverkan tas bort. Varje träd som skapas genom beskärning kandiderar sedan till posten som det slutliga trädet. Metoden för beskäring utförs med hjälp av en algoritm innehållandes den så kallade komplexitetsparametern, vilken specificerar hur mycket noggrannhet en förgrening behöver tillföra för att kunna motivera den ökade komplexiteten. Då beskärningen avslutats så väljs ett träd ut som passar, men inte överanpassar, data [53].

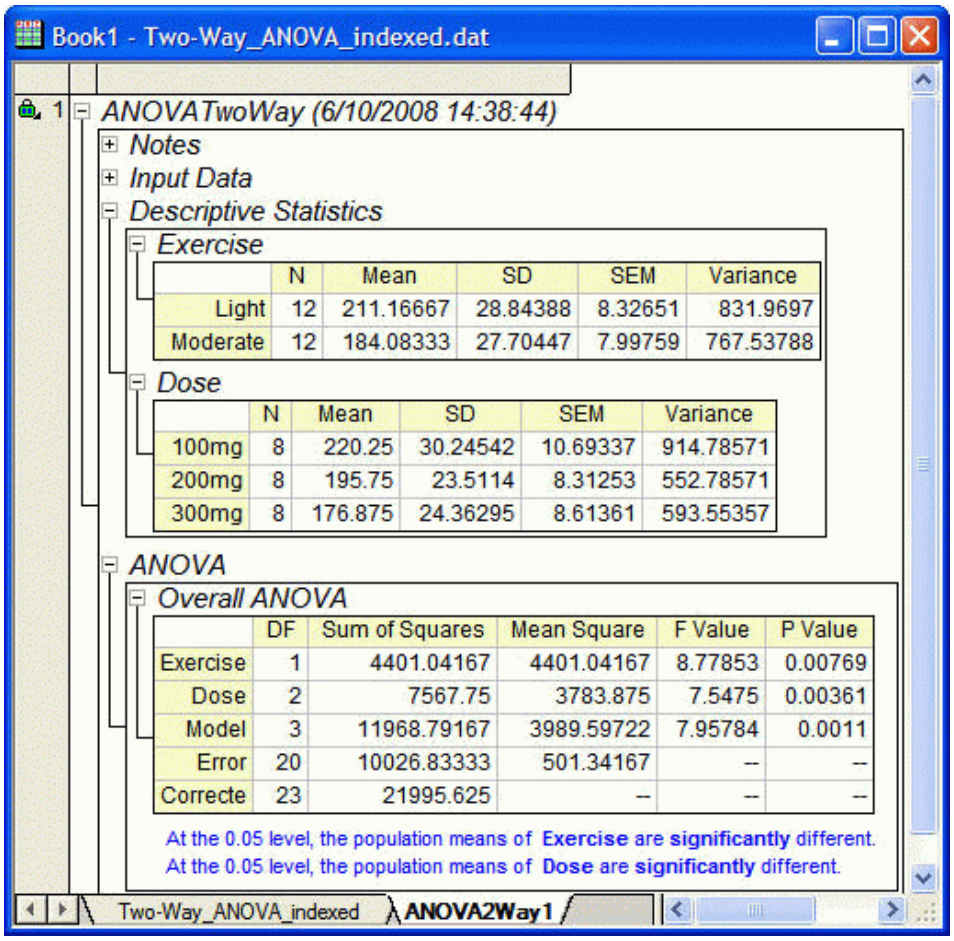
3.1.5 Variansanalys (ANOVA)

3.1.5.1 Problembeskrivning

Som bekant kan *t*-test användas för att testa nollhypotesen om två populationsmedelvärden är lika. Ofta finns det dock intresse av att jämföra fler än två populationsmedelvärden med varandra för att se om något av dessa avviker. Den typ av test som normalt används för att testa om tre (eller fler) populationsmedelvärden kan antas vara lika är variansanalys (eng. ANalysis Of VAriance, ANOVA) [28]. ANOVA är en undersökningsmetod med vilken

medelvärdenas variation studeras och metoden används alltså när syftet är studera fler än två tillstånd [30].

Genom att utföra en variansanalys kan vi avgöra huruvida den förändring som observerats kommer sig av en faktisk förändring av värdena eller om den orsakats av slumpmässigt brus. Testet resulterar i en sannolikhet för att de observerade värdeförändringarna är helt slumpmässiga, och om skillnaderna visar sig vara signifikanta så indikerar det att den resulterande slutsatsen (dvs. att medelvärdena skiljer sig åt) mest troligt inte har uppkommit enbart av slumpen. Med andra ord så är det mest troligt att det är verkliga skillnader som observerats. En envägs-ANOVA utförs då endast en kontrollerad faktor finns, medan en flervägs-ANOVA utförs för fler än två faktorer [54][55]. Genom att använda någon statistisk programvara kan information om systemet snabbt utläsas vilket exemplifieras i figur 24.



	N	Mean	SD	SEM	Variance
Light	12	211.16667	28.84388	8.32651	831.9697
Moderate	12	184.08333	27.70447	7.99759	767.53788

	N	Mean	SD	SEM	Variance
100mg	8	220.25	30.24542	10.69337	914.78571
200mg	8	195.75	23.5114	8.31253	552.78571
300mg	8	176.875	24.36295	8.61361	593.55357

	DF	Sum of Squares	Mean Square	F Value	P Value
Exercise	1	4401.04167	4401.04167	8.77853	0.00769
Dose	2	7567.75	3783.875	7.5475	0.00361
Model	3	11968.79167	3989.59722	7.95784	0.0011
Error	20	10026.83333	501.34167	--	--
Correcte	23	21995.625	--	--	--

Figur 24. Resultat från en tvåvägs-ANOVA [56].

Variansanalys kan användas till mycket mer än att bara testa nollhypotesen att ett antal populationer har samma medelvärde, exempelvis kan test utföras för att se om andra faktorer än bara den faktorn som definierats har betydelse för utfallet, eller om interaktionen mellan två olika faktorer har betydelse för utfallet [28].

3.1.5.2 Metodbeskrivning

Variansanalys utgår från antagandet att de populationer som stickproven kommer från är normalfördelade med lika varianser. Vid en envägs-ANOVA testas just nollhypotesen H_0 att alla medelvärden är lika. För att testa nollhypotesen används varianserna, närmare bestämt kvoten mellan två oberoende stickprovsvarianser. Denna kvot kommer att följa F-fördelningen och utgör vår testvariabel. Det kritiska området av F-fördelningen i en variansanalys är fördelningens högra svans och en stor F-kvot indikerar att resultatet

osannolikt har orsakats av enbart slumpen. För att se mer exakt hur osannolikt det är kan F-fördelningen användas för att få fram p-värdet [57].

Om envägs-ANOVA:n resulterar i att H_0 förkastas så blir följdfrågan ofta på vilket sätt som de aktuella populationernas väntevärden skiljer sig åt. Eftersom variansanalysen inkluderar all data på en och samma gång måste också uppföljande test göra detsamma för att vara giltiga – därmed är inte t-tester ett alternativ. Däremot kan man exempelvis använda sig av Tukeys test eller Bonferronimetoden [28]. Visar variansanalysen på en signifikant effekt så måste detta dock analyseras utifrån vad vi hoppas uppnå med resultatet – det enda vi hittills visat är ju faktiskt att sannolikheten för att resultaten har uppkommit enbart av en slump är liten [55].

Metoden för flervägs-ANOVA är snarlik den för den enkla variansanalysen. Samma antaganden gäller som för envägs-ANOVA och brustermerna antas också vara oberoende. Snarlika nollhypoteser används också för varje oberoende variabel, nämligen att alla medelvärden är lika och också att det inte existerar någon interaktion mellan några av de oberoende variablerna och inte heller mellan kombinationer av dem [20][58].

3.1.6 Lådagram

3.1.6.1 Problembeskrivning

Ett lådagram (eng. boxplot) är ett slags diagram som används för att på ett snabbt och intuitivt sätt visa fördelningen hos en kvantitativ datamängd [45][59]. Teknikens robusthet mot eventuella anomalier gör den speciellt användbar inom tillämpningsområdet för data farming. Genom att inkludera medianen i diagrammet ges en uppfattning om datamängdens centraltendens, och spridningen på data reflekteras av det så kallade interkvartila intervallet (eng. interquartile range, IQR), vilket är det område som rymmer 50% av data. Spridningsdiagram har ett flertal olika användningsområden, exempelvis som ett inledande steg vid analyser av flerdimensionella datamängder där inledande kunskap om varje variabel behövs innan multivariata analystekniker tar vid [60]. Spridningsdiagram kan helt enkelt användas för att ”lära känna” den aktuella datamängden och resultatet kan utgöra en utgångspunkt för utformningen av den fortsatta analysen.

3.1.6.2 Metodbeskrivning

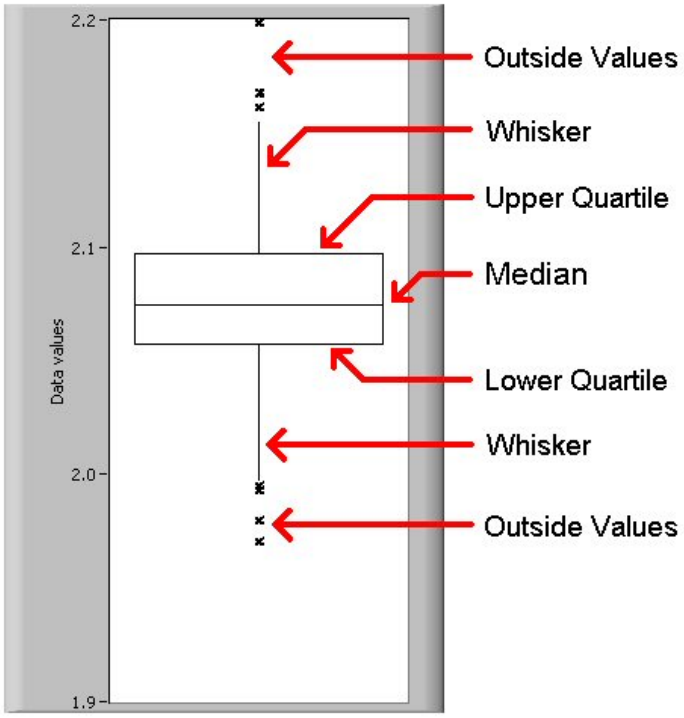
Ett lådagram delar in datamängden i kvartiler där själva ”lådan” som bygger upp lådagrammet räcker från den första kvartilen till den tredje. Inuti lådan dras en horisontell eller vertikal (beror på hur man valt att orientera lådagrammet) linje vid andra kvartilen: medianen hos datamängden [59].

Lådagrammet avläses på följande vis [59]:

- medianen indikeras genom den horisontella (vertikala) linje som går ned genom mitten av lådan,
- två vanliga mått på variabiliteten eller spridningen hos datamängden visas:
  - räckvidd (eng. range),
  - interquartile range (IQR).

En anomali definieras i lådagrammet som en datapunkt som ligger bortom första respektive tredje kvartilen med mer än 1,5 ggr IQR utanför den översta respektive nedersta kvartilen [45].

Som visas i figur 25 sträcker två vertikala linjer, kallade morrhår (eng. whiskers) ut sig från över- och undersidan av lådan. Ett sträcker sig från första kvartilen till den datapunkt med minst värde (som inte är en anomali) i datamängden medan det andra sträcker sig från tredje kvartilen till den största datapunkten. Eventuella anomalier ritas ut separat som punkter i diagrammet [59].



Figur 25. Ett lådagram med dess olika beståndsdelar [61].

Slutligen tillhandahåller lådagram också ofta information om formen (eng. shape) hos datamängden – t.ex. om datamängden är symmetrisk eller kanske uppvisar skevhet åt något håll. Denna information fås genom att medianens placering på lådan studeras: en centrerad median indikerar symmetri medan en median förskjuten ut mot kanterna indikerar skevhet [59]. Vid jämförelse av flera lådagram kan insikter gällande varians också utvinnas – om lådorna är av väldigt olika storlekar indikerar det olika varianser [62].

3.1.7 Principalkomponentsanalys (PCA)

3.1.7.1 Problembeskrivning

PCA (eng. Principal Component Analysis) är en enkel icke-parametrisk metod som används för att extrahera värdefull information från svårgreppbara datamängder [63]. PCA är ett sätt att identifiera mönster i data och utrycka data på ett sådant sätt att såväl likheter som skillnader belyses [64]. Ofta är dessa datamängder av ett högt antal dimensioner och då flerdimensionella rum är svåra att visualisera så används PCA främst för att reducera dimensionaliteten hos en datamängd ned till mer hanterbara två eller tre dimensioner. Genom detta hoppas man på ett tydligare sätt kunna se de underliggande strukturer och mönster som tidigare varit dolda. Kortfattat kan sägas att PCA lägger ihop variationen i en korrelerad mängd innehållandes flera attribut så att resultatet blir en uppsättning av okorrelerade komponenter där varje komponent är en linjär kombination av de ursprungliga variablerna. Den nya basen är alltså en linjär kombination av den ursprungliga basen och är den bas som bäst återuttrycker vår brusiga ursprungliga datamängd [63].

Målet med PCA är alltså att reducera dimensionaliteten genom att extrahera det minsta antalet komponenter som svarar för den största delen av variationen i den ursprungliga flervariata datamängden och hitta mönster i datamängden utan någon större informationsförlust [64][65]. PCA tillhandahåller en slags guide till hur en komplex datamängd reduceras till en lägre dimension och eftersom PCA gör ett grundantagande om linjäritet så reduceras problemet till att hitta ett passande basbyte [63].

3.1.7.2 Metodbeskrivning

Vi vill visualisera våra datapunkter vilket kan åstadkommas genom att projicera ned punkterna till två eller tre dimensioner. Utmaningen är att avgöra vilket underrum, eller axlarna, som vi projicerar på. Den riktning vi projicerar på avgör nämligen vad det är vi ser i slutändan, varpå vi vill hitta den riktning vars projektioner är mest utspridda vilket i sin tur kan göras genom variansen hos projektionerna [66]. PCA antar att riktningarna med de största varianserna är de viktigaste, med andra ord – de är mest principala [63].

I PCA extraheras okorrelerade principalkomponenter genom linjära transformationer av de ursprungliga variablerna. Detta resulterar i att de första principalkomponenterna innehåller den största delen av variationen i den ursprungliga datamängden. Efter PCA kan de första principalkomponenterna försöka tolkas i termer av de ursprungliga variablerna och därigenom kan förhoppningsvis en större förståelse för datamängden fås [65].

Det olika stegen i PCA kan kortfattat beskrivas som följer [64][66]:

1. Datainsamling.
2. Medelvärdessubtrahering. Väntevärdet från var och en av datadimensionerna subtraheras, vilket resulterar i en datamängd vars medelvärde är noll,
3. Beräkning av kovariansmatrisen.
4. Hitta egenvektorerna till kovariansmatrisen. Egenvektorerna skall vara normaliserade. Genom att hitta dessa vektorer är det möjligt att extrahera linjer som karaktäriserar data. Vi vill sedan transformera data så att de uttrycks i termer av dessa linjer.
5. Beräkna egenvektorernas egenvärden. Ordna egenvärdena i storleksordning. Det är i detta steg som begreppet datakomprimering och reducerad dimensionalitet blir aktuellt. Den av vektorerna som har det största egenvärdet kallas nämligen datamängdens *första principalkomponent*. Det är det mest signifikanta förhållandet mellan datadimensionerna och det största egenvärdet är värdet hos den (maximala) variansen av projiceringarna på den första principalkomponenten. Den andra *principalkomponenten* har näst störst egenvärde, osv. Det antal principalkomponenter vi väljer bestämmer den nya dimensionaliteten.

3.1.8 Glyfer

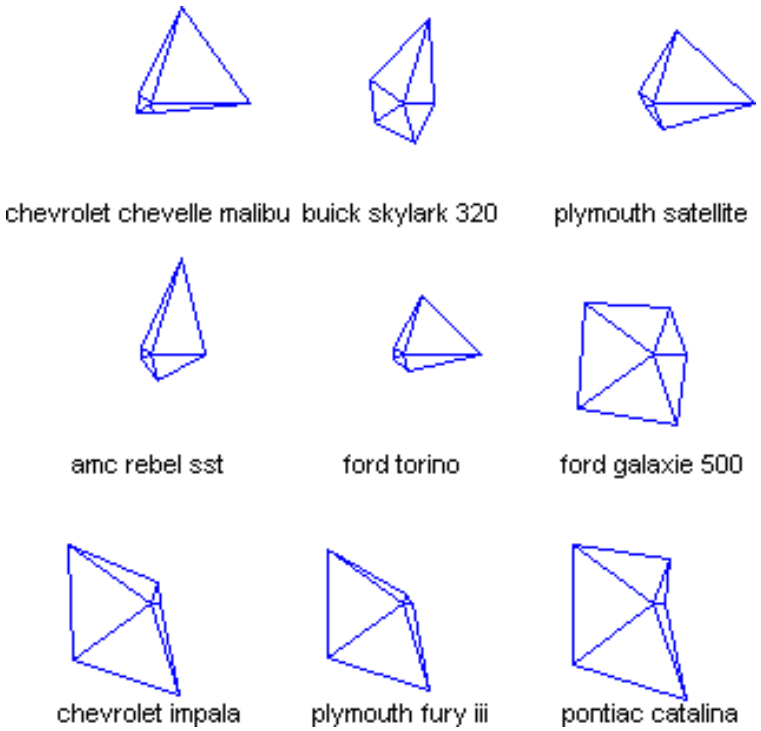
3.1.8.1 Problembeskrivning

En användbar teknik för att visa flerdimensionella data är så kallade glyfer (eng. glyphs), vilka är små ikoner som ändrar utseende utefter de data de representerar. En glyf ritas för varje flerdimensionell datapunkt. Genom att belysa gemensamheter och skillnader i glyfernas utseenden kan tekniken visa viktiga strukturer i den ursprungliga datamängden. Två populära varianter av glyfer är stjärnglyfer (eng. star glyphs eller star plots) och Chernoff-ansikten (eng. Chernoff faces) [67][68].

3.1.8.2 Metodbeskrivning

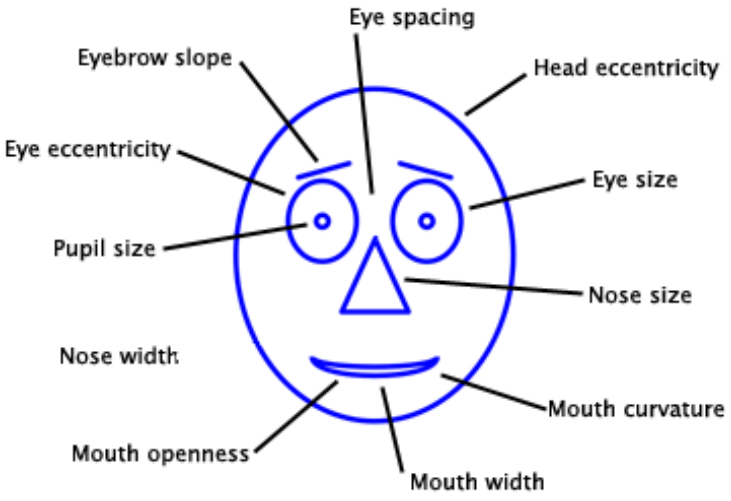
Ett stjärndiagram består av en punkt med ”ekrar” som utgår från punkten. Varje dimension i datamängden motsvarar en eker och längden på varje eker representerar variabelns storlek jämfört med maxvärdet av alla variablerna. Stjärnglyfer är mycket effektiva att använda för att avgöra var data börjar klustra sig samt var eventuella anomalier återfinns, dock kan diagrammet bli svåröverblickligt vid en alltför stor datamängd [67][68].

I en stjärnglyf är vinkelavståendet mellan varje eker detsamma. En stjärnglyf har två dimensioner: en kvantifierbar dimension och en kategorisk dimension. Den kvantifierbara dimensionen är ett enda värde och storleken på detta värde representeras av längden hos en eker. Det kvantifierbara värdet grupperas av kategoriska data [68]. I figur 26 presenteras ett exempel på hur stjärnglyfer kan se ut.



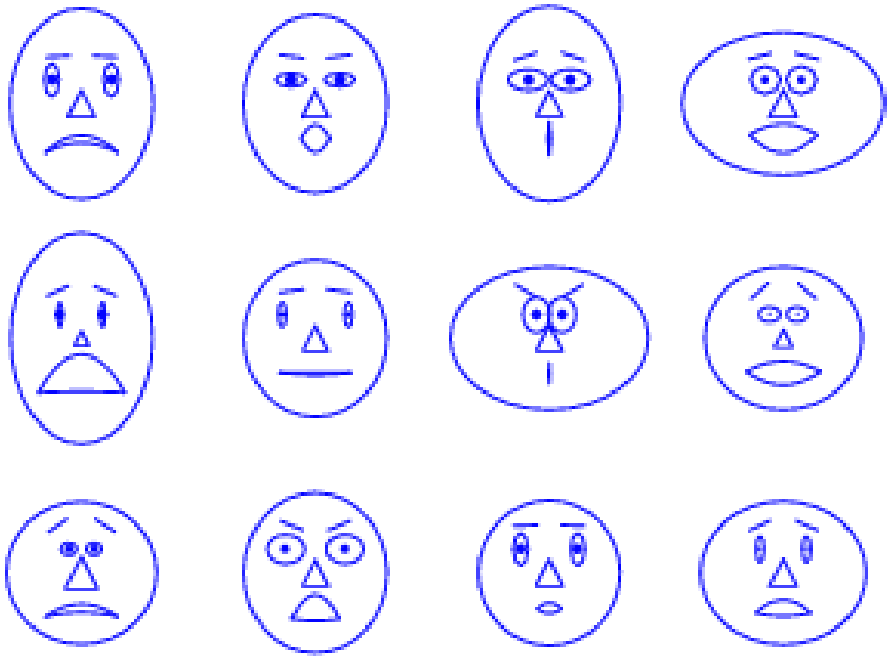
Figur 26. Exempel på stjärnglyfer där glyferna representerar olika bilmodeller [69].

Chernoff-ansikten är mer komplicerade glyfer, i vilka varje datapunkt representeras med en stiliserad bild av ett människoansikte. Storlek och form på, samt avståndet mellan de olika ansiktsdelarna (t.ex. ögonen) representerar betydelsen hos olika variabler. Styrkan hos Chernoff-ansikten är att tekniken utnyttjar att människor är väldigt duktiga på att särskilja och känna igen människoansikten [67].



Figur 27. Beskrivning av de olika ansiktsdrag som är av betydelse för Chernoff-ansikten [70].

För att skapa Chernoff-ansikten så låter man helt enkelt inkludera så många ansiktsdrag som ens datamängd har variabler. Figur 28 visar Chernoff-ansikten som skapats genom att använda tio stycken ansiktsdrag som alla har antagit olika värden för varje ansikte [71].



Figur 28. Chernoff-ansikten med tio olika ansiktsdrag [71].

3.1.9 Parallella koordinatplottar

3.1.9.1 Problembeskrivning

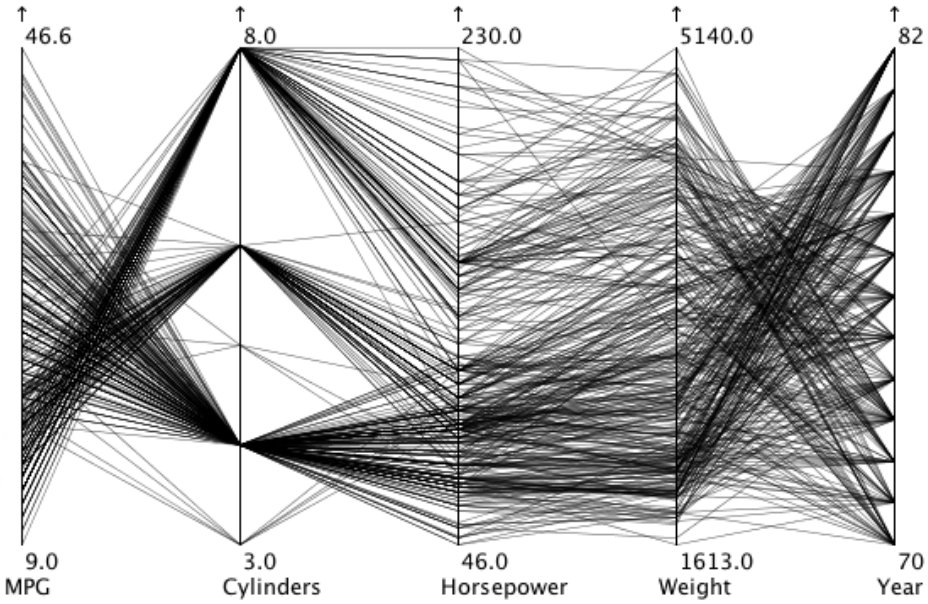
Parallella koordinatplottar är en teknik som används för att visualisera flerdimensionella data. Till skillnad från andra flerdimensionella visualiseringstekniker såsom exempelvis stjärnglyfer och Chernoff-ansikten så kan parallella koordinatdiagram göra det litet enklare att identifiera grupper eller identifiera relationer mellan variabler. Dock är stjärnglyfer och Chernoff-ansikten effektivare när det gäller att hitta enheter som sticker ut från resten [45].

Den stora behållningen med parallella koordinater är kanske främst teknikens förmåga att belysa viktiga multivariata mönster och möjliggöra jämförelser när de används interaktivt för analys. Tekniken är också speciellt användbar för att identifiera villkor som är starkt korrelerade med ett specifikt utfall eftersom parallella koordinater kan åskådliggöra korrelation mellan multipla variabler [72].

Trots att tekniken initialt kan ge ett väldigt komplext intryck utgör parallella koordinatplottar ett kraftfullt verktyg för att öka vår systemförståelse. När så pass mycket data visas på en och samma gång kan underlaget förmodligen inte användas för att utforska detaljerna men istället erbjuder tekniken insikter om dominanta mönster och anomalier och möjlighet att få en helhetsbild över utdata [72].

3.1.9.2 Metodbeskrivning

För att skapa en parallella koordinaterplot placeras multipla axlar parallellt med varandra där toppen representerar en variabels maximum och botten minimet. För varje enhet dras en linje från vänster till höger (om man väljer att orientera diagrammet på det sättet) och linjen rör sig i höjdled beroende på enhetens värde [45]. Linjernas upp- och nedåtlutningar indikerar alltså den förändring som sker med tiden från ett värde till det nästa. Dock indikerar inte linjerna i sig själva förändring utan varje linje i diagrammet binder samman en serie av värden, där varje värde är associerat med en egen variabel, och mäter flera aspekter av någonting, exempelvis av en person [72]. Slutligen bildar dessa axlar och linjer ett diagram såsom det som visas i figur 29.



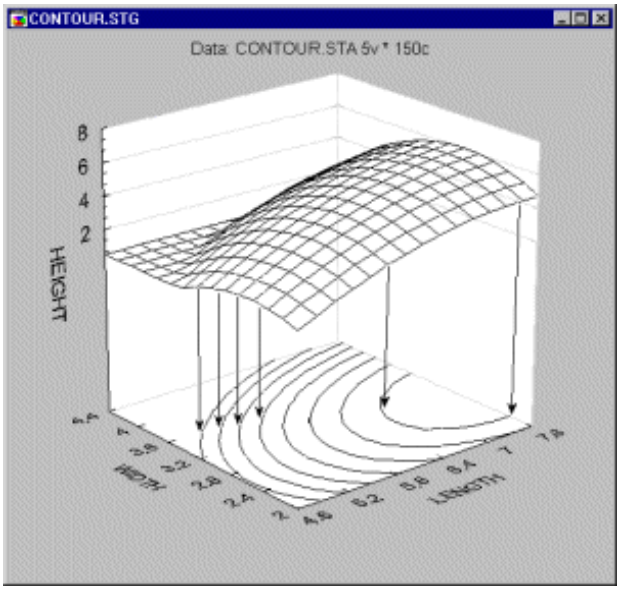
Figur 29. Parallella koordinater som beskriver olika bilmodeller utifrån fem olika variabler [73].

Om det är möjligt att gruppera data i kluster så kan det vara bra att sedan dela in dessa så att kluster av liknande data hamnar i separata grafer. På så sätt underlättas analysarbetet då det blir enklare att fokusera på en specifik grupp skild från de andra, samt för att kunna jämföra klustrens olika multivariata profiler. Jämförelser kan också göras mellan flera uppsättningar av kombinationer av olika faktorer [72].

3.1.10 Konturplot

3.1.10.1 Problembeskrivning

Konturplottar (eng. contour plots) är en teknik som används för att grafiskt representera förhållandena mellan tre numeriska variabler i två dimensioner och ger insikt i hur en responsvariabel förändras som en funktion av två förklarande variabler. I figur 30 visas hur den tredimensionella ytan projiceras ned på ett plan.

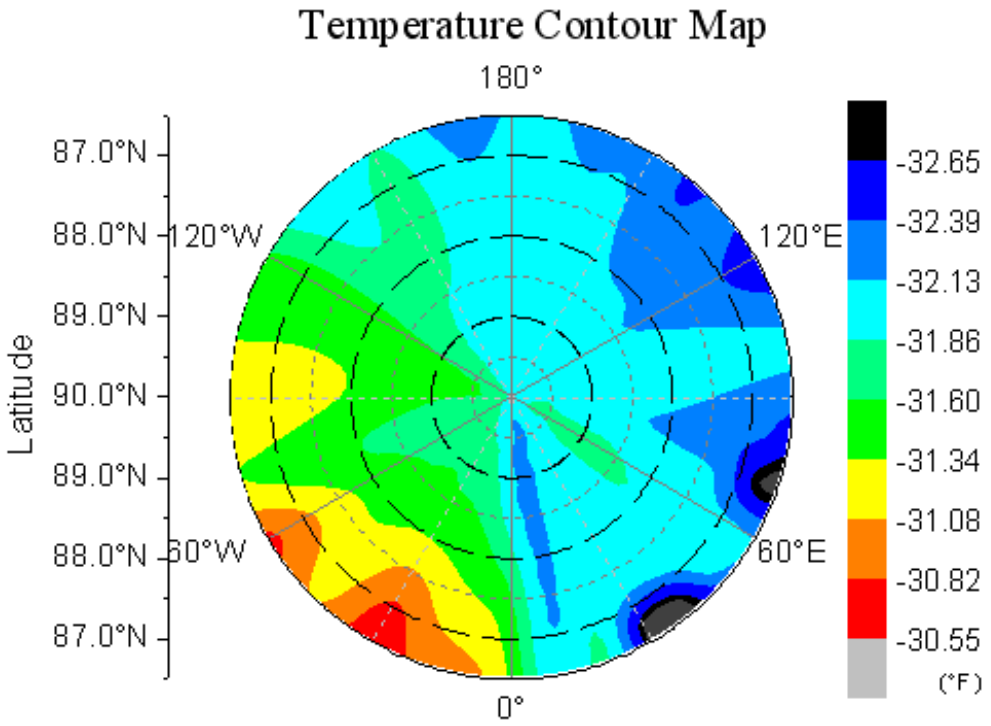


Figur 30. Konturplot som visar projektion från tre ned på två dimensioner [74].

Två av variablerna motsvarar *X*- respektive *Y*-axeln medan den tredje variabeln *Z* motsvarar de så kallade konturnivåerna. Konturnivåerna ritas ut som kurvor och arean mellan dessa kurvor kan färgkodas för att indikera interpolerade värden. För stora datamängder kan en konturplot vara ett viktigt första steg till att skapa förståelse för data [20][75][76].

3.1.10.2 Metodbeskrivning

En konturplot för en funktion $z = f(x,y)$, visar (x,y)-planet – alltså funktionens domän. För varje z-värde ritas en linje, kallad isobar (eng. isobar) ut på planet. Ett exempel skulle kunna vara en väderkarta där x- och y-värdena skulle ge platsens läge medan z (som alltså är en funktion av de två andra variablerna) skulle indikera temperaturen på den platsen. Genom att studera den resulterande konturplotten kan insikter fås gällande var funktionen förändras snabbt respektive långsamt, när den befinner sig vid maximum resp. minimum och där den är konstant [76]. Konturplottar kan skapas från två sorters data: så kallad ”gridded data” lagrad i en matris eller *XYZ*-tripletter [77]. Avslutningsvis representerar figur 31 ett exempel på hur en konturplot för temperaturer kan se ut.



Figur 31. Konturplot som visualiserar temperaturfördelning [78].

3.2 Verktyg

Två lämpliga verktyg för att analysera och visualisera utdata från simuleringsmodeller är JMP[4] och MATLAB Statistics Toolbox[5]. Vi tror att JMP passar bäst för operations-analytiker då det går snabbt att komma igång med verktyget, medan MATLAB Statistics Toolbox passar bäst för systemutvecklare då man har full tillgång till egen källkod och kan bygga inbäddade lösningar.

[4] JMP (2011). SAS Institute Inc., Cary, NC. [Online] http://www.jmp.com/
[5] MATLAB Statistics Toolbox (2011). MathWorks, Natick, MA. [Online] http://www.mathworks.se/products/statistics/

4 Referenser

[1] Horne G.E, Meyer, T.E. (2005), Data farming: Discovering surprise, in Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (eds.), *Proceedings of the 2005 Winter Simulation Conference*, Orlando, FL, 4–7 December 2005, pp. 1082–1087. [Online] http://www.informs-sim.org/wsc05papers/130.pdf

[2] K computer. [Online] http://www.top500.org/

[3] Russell, J. (1926), Field experiments: How they are made and what they are, *Journal of the Ministry of Agriculture of Great Britain* **32**:989–1001.

[4] Fisher, R.A. (1926), The arrangement of field experiments, *Journal of the Ministry of Agriculture of Great Britain* **33**:503–513.

[5] Kleijnen, J.P.C., Sanchez, S.M., Lucas, T.W., Cioppa, T.M. (2005), A User's Guide to the Brave New World of Designing Simulation Experiments, *INFORMS Journal on Computing* **17**(3):263–289. [Online] http://harvest.nps.edu/papers/UserGuideSimExpts.pdf

[6] Sanchez, S.M. (2007), Work smarter, not harder: Guidelines for designing simulation experiments, in Henderson, S.G., Biller, B., Hsieh, M.-H., Shortle, J., Tew, J.D., Barton, R.R. (eds.), *Proceedings of the 2007 Winter Simulation Conference*, Washington, DC, 9–12 December 2007. IEEE, Piscataway, NJ, pp. 84–94. [Online] http://www.informs-sim.org/wsc07papers/010.pdf

[7] Box, G.E.P., Hunter, W.G., Hunter, J.S. (1978), *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*. John Wiley & Sons, New York, NY.

[8] Sanchez, S.M., Sanchez, P.J. (2005), Very large fractional factorials and central composite designs, *ACM Transactions on Modeling and Computer Simulation* **15**(4):362–377. [Online] http://harvest.nps.edu/papers/SanchezSanchezACM_TOMACS_05.pdf

[9] Bettonvil, B., Kleijnen, J.P.C. (1996), Searching for important factors in simulation models with many factors: Sequential bifurcation, *European Journal of Operational Research* **96**(1):180–194. [Online] http://dx.doi.org/10.1016/S0377-2217(96)00156-7

[10] Cioppa, T.M., Lucas, T.W. (2007), Efficient nearly orthogonal and space-filling Latin hypercubes, *Technometrics* 49(1):45-55. [Online] http://harvest.nps.edu/papers/Cioppa.Lucas.pdf

[11] Hernandez, A.S. (2008), Breaking barriers to design dimension in nearly orthogonal Latin hypercubes, Ph.D. thesis. Naval Postgraduate School, Monterey, CA. [Online] http://www.dtic.mil/dtic/tr/fulltext/u2/a494168.pdf

[12] Hernandez, A.S., Lucas, T.W., Carlyle, M. (2011), Constructing nearly orthogonal Latin hypercubes for any nonsaturated run-variable combination, Working Paper, Naval Postgraduate School, Monterey, CA.

[13] Viana, F.A.C., Venter, G., Balabanov, V. (2010), An algorithm for fast optimal Latin hypercube design of experiments, *International Journal for Numerical Methods in Engineering* **82**(2):135–156. [Online] http://dx.doi.org/10.1002/nme.2750

[14] Sanchez, S.M., Wan, H. (2009), Better than a petaflop: The power of efficient experimental design, in Rossetti, M.D., Hill, R.R, Johansson, B., Dunkin, A. Ingalls, R.G. (eds.), *Proceedings of the 2009 Winter Simulation Conference*, Austin, TX, 13–16 December 2009. IEEE, Piscataway, NJ, pp. 60–74. [Online] http://www.informs-sim.org/wsc09papers/007.pdf

[15] Vieira Jr., H., Sanchez, S., Kienitz, K.H., Belderrain, M.C.N. (2011), Generating and improving orthogonal designs by using mixed integer programming, *European Journal of Operational Research* **215**(3):629–638. [Online] http://dx.doi.org/10.1016/j.ejor.2011.07.005

[16] Vieira Jr., H., Sanchez, S.M., Kienitz, K.H. (2011), Improved efficient, nearly orthogonal, nearly balanced mixed designs, in Jain, S., Creasey, R.R., Himmelspach, J., White, K.P., Fu, M. (eds.), *Proceedings of the 2011 Winter Simulation Conference*, Phoenix, AZ, 11–14 December 2011. IEEE, Piscataway, NJ, pp. 3605–3616. [Online] http://www.informs-sim.org/wsc11papers/320.pdf

[17] Scatterplot (2011). [Online] http://www.stat.yale.edu/Courses/1997-98/101/scatter.htm

[18] Scatter plot matrix (2011). [Online] http://trellischarts.com/documentation/scatter-plot-matrix

[19] Bright, D.B., Williams, S.E. (2005), Scatterplot matrices, in *Encyclopedia of Statistics in Behavioral Science*, Vol. 4. John Wiley & Sons, New York, NY, pp. 1794–1795.

[20] e-handbook of statistical methods (2011), NIST/SEMATECH. [Online] http://www.itl.nist.gov/div898/handbook

[21] Scatter plot matrix (2011). [Online] http://sites.stat.psu.edu/~lsimon/stat501wc/sp05/minitab/scatter_plot_matrix.html

[22] Regressionsanalys – en introduktion (2011). [Online] http://www.bbs.hik.se/utbildning/kurssidor/statistik/filer%206-10p/regression.pdf

[23] Scatter plot (2011). [Online] http://www.netmba.com/statistics/plot/scatter/

[24] Sykes, A.O. (1993), An introduction to regression analysis, Working Paper in Law and Economics Number 020, Institute for Law and Economics, University of Chicago, IL.

[25] Strobl, C., Malley, J., Tutz, G. (2009), An introduction to recursive partitioning, Technical Report Number 55, Institut für Statistik, Ludwig-Maximilians-Universität München.

[26] Choudhury, A. (2011), Correlation and regression. [Online] http://www.experiment-resources.com/correlation-and-regression.html

[27] Least-squares regression line (2011). [Online] http://www.une.edu.au/WebStat/unit_materials/c4_descriptive_statistics/least_squares_regress.html

[28] Lantz, B. (2009), *Lär lätt! Statistik kompendium*. Ventus Publ., Holstebro.

[29] Konfidensintervall (2011). [Online] http://www.math.kth.se/matstat/gru/5b1501/V/f9.pdf

[30] Rutherford, A., (2001), *Introducing ANOVA and ANCOVA a GLM approach.* Sage Publ., Gateshead.

[31] Dallal, G. (2001), What does multiple linear regression look like? [Online] http://www.jerrydallal.com/LHSP/regpix.htm

[32] Multiple regression model (2011). [Online] http://www.unesco.org/webworld/idams/advguide/Chapt5_2.htm

[33] Multipel regressionsanalys (2011). [Online] http://www.ida.liu.se/~732G71/info/forel2_09.ppt

[34] Multiple regression (2011). [Online]
http://www.fordham.edu/economics/vinod/multiple-regression.doc

[35] Lindgren, G. (2011), Statistik för modellval och prediktion – att beskriva, förklara och förutsäga, Institutionen för matematisk statistik, Lunds universitet. [Online]
http://www.maths.lth.se/matstat/staff/georg/SMHI/Kurs_del_2_4.pdf

[36] Johansson, T., Carlsson, O. (2003), Tågtrafiken – punktlighet och förseningar, FOU rapport T 2003:2.1. Banverket Trafik, Borlänge. [Online]
http://www4.banverket.se/raildokuffe/pdf/MN0023.pdf

[37] Stepwisefit (2011). [Online]
http://www.mathworks.se/help/toolbox/stats/stepwisefit.html

[38] Stepwise regression (2011). [Online]
http://www.investopedia.com/terms/s/stepwise-regression.asp#axzz1iO1mF5w5

[39] Anderson, D.R., Sweeney, D.J., Williams, T.A., Freeman, J., Shoesmith, E. (2010), *Statistics for Business and Economics*. Thomson South-Western, Mason, OH.

[40] Hawkins, D.M. (2004), The Problem of Overfitting, *Journal of Chemical Information and Computer Sciences* **44**(1):1−12. [Online]
http://www.cbs.dtu.dk/courses/27618.chemo/overfitting.pdf

[41] Häggström, J. (2005), Loess och korsvalidering – Val av utjämningsparameter. D-uppsats, Statistiska Institutionen, Umeå Universitet. [Online]
http://www.stat.umu.se/kursweb/vt05/stad05mom3/?download=JennyH.pdf

[42] Fox, J. (2005), Introduction to nonparametric regression, Department of Politics and International Relations, McMaster University, Hamilton, Ontario. [Online]
http://socserv.mcmaster.ca/jfox/Courses/Oxford-2005/slides-handout.pdf

[43] Cleveland, W.S., Devlin, S.J. (1988), Locally weighted regression: An approach to regression analysis by local fitting, *Journal of the American Statistical Association* **83**(403):596-610.

[44] Glynn, E.F. (2005), Loess smoothing and data imputation. [Online]
http://research.stowers-institute.org/efg/R/Statistics/loess.htm

[45] Yau, N. (2011), *Visualize This*. John Wiley & Sons, New York, NY.

[46] Timofeev, R. (2004), Classification and regression trees (CART) theory and applications, Rapport 188778, Center of Applied Statistics and Economics, Humboldt University, Berlin.

[47] Loh, W. (2008), Classification and regression tree methods, in *Encyclopedia of Statistics in Quality and Reliability*. John Wiley and Sons, New York, NY, pp. 315−323.

[48] Classification and regression trees (2011). [Online]
http://www.dtreg.com/classregress.htm

[49] Richkowski, D.M. (2008), Performance metrics and analysis of small unmanned ground vehicles (SUGVS) in building clearing operations. Master Theisis, Naval Postgraduate School, Monterey, CA.

[50] Lazzati, N. (2011), Quasiconcave and Pseudoconcave Functions, Mathematics for Economics (Part I). University of Arizona,Tucson, AZ. [Online]
http://www.u.arizona.edu/~nlazzati/Courses/Math519/Notes/Note%2010.pdf

[51] Gustafsson, H., Larsson, A. (2001), Hur kan beslutstödsprocessen stödjas av data mining?, Institutionen för Informatik, Göteborgs Universitet. [Online]
http://gupea.ub.gu.se/bitstream/2077/1411/1/33687-IA7300gustafssonlarsson.pdf

[52] Classification and regression trees (CART) (2011). [Online] http://www.cems.uwe.ac.uk/~rblawton/classification%20and%20regression%20trees.ppt

[53] Lewis, R.J. (2000), An introduction to classification and regression tree (CART), in *Proceedings of the 2000 Annual Meeting of the Society for Academic Emergency Medicine*, San Francisco, CA, 22–25 maj 2000, 14 pp. [Online] http://crocea.mednet.ucla.edu/research/annot/cart/doc/lewis2000.pdf

[54] Variansanalys (2011). [Online] http://www.ollevejde.se/statistikord/variansanalys.htm

[55] Dataövning 1 – ANOVA och MLR (2011). [Online] http://www.studentportalen.uu.se

[56] Origin: Two-way ANOVA (2011). [Online] http://www.originlab.com/index.aspx?go=Products/Origin/Statistics/ANOVA&pid=123

[57] Labreflektion datorövning 1: ANOVA & MLR (2011). [Online] http://www.studentportalen.uu.se

[58] Stats: Two-way ANOVA (2011). [Online] http://people.richland.edu/james/lecture/m170/ch13-2wy.html

[59] Boxplots (aka box and whisker plots) (2011). [Online] http://stattrek.com/ap-statistics-1/boxplot.aspx

[60] Massart, D.L., Smeyers-Verbeke, J., Capron, X., Schleser, K. (2005) Visual presentation of data by means of box plots, *LCGC Europe* **18**(4):215–218. [Online] http://www.lcgceurope.com/lcgceurope/data/articlestandard/lcgceurope/132005/152912/article.pdf

[61] Boxplots and stem-and-leaf displays (2011). [Online] http://zone.ni.com/devzone/cda/tut/p/id/3047

[62] Faraway, J.J. (2002), Practical regression and Anova using R. [Online] http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf

[63] Shlens, J. (2005), A tutorial on principal component analysis. [Online] http://www.cs.cmu.edu/~elaw/papers/pca.pdf

[64] Smith, L.I. (2002), A tutorial on principal components analysis. [Online] http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[65] Fernandez, G. (2011), Principal component analysis. [Online] http://www.ag.unr.edu/saito/classes/ers701/pca2.pdf

[66] Principal component analysis (PCA) (2011). [Online] http://www.weizmann.ac.il/home/fedomany/Bioinfo05/lecture3.pdf

[67] Spears, W.M. (1999), An overview of multidimensional visualization techniques, in Collins, T.D. (ed.), *Proceedings of the Evolutionary Computation Visualization Workshop,* Orlando, FL, pp. 104–105. [Online] http://www2.iath.virginia.edu/time/readings/visualization-representation/visualization-overview.pdf

[68] Rusu, A., Santiago, C., Crowell, A., Thomas, E. (2009), Enhanced star glyphs for multiple-source data analysis, in *Proceedings of the 13th International Conference on Information Visualisation*, Barcelona, Spain, 15–17 juli 2009. IEEE, Piscataway, NJ, pp 183–190. [Online] http://acy.tc.faa.gov/cpat/docs/IV09_StarGlyphDataAnalysis.pdf

[69] Visualizing multivariate data (2011). [Online] http://www.mathworks.se/products/statistics/demos.html?file=/products/demos/shipping/stats/mvplotdemo.html

[70] Chernoff faces (2011). [Online] http://kspark.kaist.ac.kr/Human%20Engineering.files/Chernoff/Chernoff%20Faces.htm

[71] Chernoff face (2011). [Online] http://mathworld.wolfram.com/ChernoffFace.html

[72] Few, S. (2006), Multivariate analysis using parallel coordinates. [Online] http://www.perceptualedge.com/articles/b-eye/parallel_coordinates.pdf

[73] Kosara, R. (2010), Parallel coordinates. [Online] http://eagereyes.org/techniques/parallel-coordinates

[74] INSIGHT user's guide, ver. 8 (1999), SAS Institute Inc., Cary, NC. [Online] http://www.okstate.edu/sas/v8/saspdf/insight/chap36.pdf

[75] Callahan, J., Cox, D.A., Hoffman, K.R., O'She,a D., Pollatsek, H., Senechal, L. (1995), Functions of several variables, in *Calculus in Context*. W. H. Freeman Publ., New York, NY, pp. 28–32. [Online] http://www.math.smith.edu/~rhaas/m114-00/chp6_3D.pdf

[76] Igor pro user's guide, contour plots (2011). [Online] http://www.wavemetrics.net/doc/igorman/II-14%20Contour%20Plots.pdf

[77] Contour plot (2011). [Online] http://www.statsoft.com/textbook/statistics-glossary/c/button/c/

[78] Contour graphs (2011). [Online] http://www.originlab.com/index.aspx?go=Products/Origin/Graphing/Contour