Python and the Holy Grail: Developing Superior Data analysis methods

aka. DSDs with STAT-EASE 360

Andrew Macpherson
Managing Director,
Prism Training & Consultancy Ltd

Describing our Services and Details

- **Prism Training & Consultancy Limited: www.prismtc.co.uk**

- Independent company offering statistical training, consultancy and software services
- Founded in 2000
- Based in Cambridge, UK
- Consists of a small team of statisticians, trainers, programmers, scientists etc.
- 150+ global customers across multiple industries

- **Stat-Ease Reseller for UK & Ireland since 2013**

Declaring Some Deliverables

- **In this talk, we hope to illustrate how Stat-Ease's latest software allows us to explore beyond standard DoE tools, and create bespoke solutions for one-off and/or routine use**

- We will briefly revisit topics covered by Dr. Paul Nelson, our Technical Director, at Stat-Ease's Paris 2018 DoE Summit: "Practical (Real Life) Implementation of DSDs"

- Through Python, we will take you on a Quest for the Holy Grail of DSDs: an automated tool that appropriately identifies and fits the best possible model to our data!

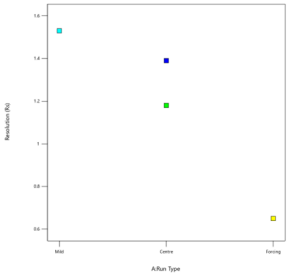
Defining Subsequent Discussion

- Describing our Services and Details
- Declaring Some Deliverables
- DX / SE360 Differences
- Detour into Software Details
- Decidedly Suspicious Disclaimers
- Design Summary & Discussion
- Defining Sample Data
- Debating Standard Design tools
- Demanding Specialised Decisions
- Delicately Splitting our Data
- Daydreaming about Software Development
- Diving into a Speedy Demo

DX / SE360 Differences

DESIGN-EXPERT

- Design-Expert provides us with plenty of user-friendly DoE functionality; more than we're likely to need in the vast majority of cases.
- However, we may occasionally want something niche / non-standard, beyond the everyday DoE toolkit…
  - For example, we sometimes recommend running a simple "Scoping Design" as an exploratory first step in sequential DoE.
  - To simplify the creation of these designs, we developed a free online tool that allows users to interactively build their own *.dxpx* files: www.prismtc.co.uk/resources/free-tools/scoping-design-builder



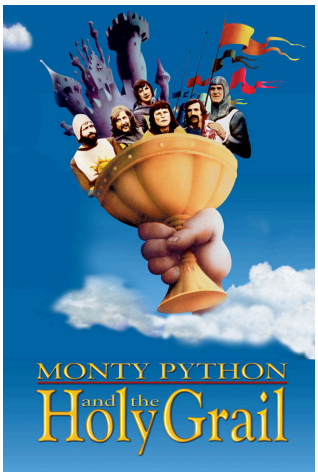
- **Stat-Ease 360 supercharges DX through "power user" functionality, including Python scripting for automation and extensibility within the Stat-Ease environment!**

STAT-EASE 360

5

Detour into Software Details

- About Python
  - Popular!
  - Open source
  - Readable and therefore (relatively) user friendly
  - Powerful and extensible
  - Named after Monty Python!
- More info at python.org



Decidedly Suspicious Disclaimers

- Warning: this DSD / Python talk is not presented by a proper statistician, nor a good Python programmer.
  - My interest is practical problem solving: helping people to find the sweet spot between science, statistics and software makes me happy!
  - … and if I can hack together a Python tool, then so can you!
- Our Python demo app has been developed solely as a worked example for this talk; we want to show how SE360's Python integration can provide enormous flexibility.
- **Disclaimers aside, we hope that our talk might inspire you to exploit SE360's extensibility for your own benefit!!**

Design Summary & Discussion

- **Definitive Screening Designs** were introduced in 2011 by Brad Jones and Chris Nachtsheim
  - For more info on DSDs, please read our article here!
- They are 3-level, foldover, alias-optimal designs, which offer an efficient approach to **screening** factors (i.e. estimation of main effects with *2k+1* runs)
- Plus, DSDs can sometimes fit a full RSM model… but not always…
  - We cannot fit a full RSM model to **all** of our factors, as we do not have enough degrees of freedom in this "supersaturated" design (e.g. 6 factors => 28 RSM model terms, from only 13 runs!?!)
  - However, if ≤ 3 factors are active, then the design projects down to allow us to fit a full RSM model to the chosen few! (3 factors => 10 RSM model terms)

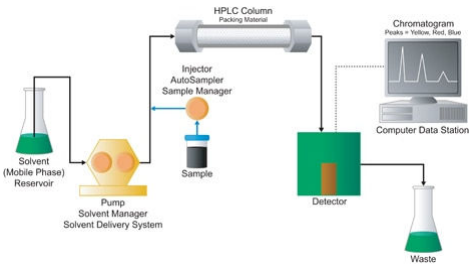
Design Summary & Discussion

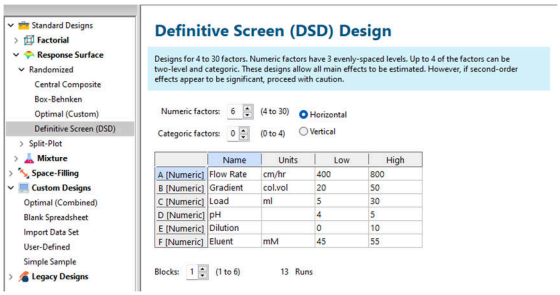
- So, how should we view DSDs??
  - If we expect full a RSM model for all factors from *2k+1* runs, then we are likely to be disappointed!
  - Instead, we suggest viewing them as ME screening designs, with a potential trick up their sleeve…
  - … and if we believe that "sparsity of factors" is likely in our chosen study, then DSDs could offer a very appealing option!

- (In this talk, we will avoid the delicate question of whether competent scientists are likely to choose to study many irrelevant factors!!)

Defining Sample Data

- Case study: Purification of Recombinant Fab in E.coli





Std	Run	Factor 1 A:Flow Rate cm/hr	Factor 2 B:Gradient col.vol	Factor 3 C:Load ml	Factor 4 D:pH	Factor 5 E:Dilution	Factor 6 F:Eluent mM	Response 1 Resolution Rs	Response 2 Flow Fab PA area	Response 3 Time min
1	3	600	50	30	5	10	55			
2	1	600	20	5	4	0	45			
3	11	800	35	5	5	10	45			
4	5	400	35	30	4	0	55			
5	2	800	20	17.5	4	10	55			
6	9	400	50	17.5	5	0	45			
7	8	800	50	5	4.5	0	55			
8	7	400	20	30	4.5	10	45			
9	13	800	50	30	4	5	45			
10	6	400	20	5	5	5	55			
11	4	800	20	30	5	0	50			
12	12	400	50	5	4	10	50			
13	10	600	35	17.5	4.5	5	50			

Debating Standard Design tools

- Design-Expert and Stat-Ease 360 offer the most widely used and flexible "Auto Select" methods: P-values, AICc, BIC & Adj R-Squared criteria, with Forward, Backward & Stepwise selection

- However, DSDs pose unique challenges that general purpose model selection tools can sometimes struggle with (e.g. supersaturated nature & partially aliased second order terms)

Fit Summary

Response 1: Resolution

	Source	Sequential p-value	Lack of Fit p-value	Adjusted R^2	Predicted R^2	
	Linear	**0.0577**		**0.6006**	**-0.0434**	**Suggested**
	2FI	0.4835		0.7874	-32.1787	**Aliased**

- **In order to fully unlock the potential advantages of DSDs, we might benefit from a bespoke solution…**

Demanding Specialised Decisions

- DSDs contain foldover pairs of runs, and the structure makes MEs orthogonal to second order terms.
  - Jones & Nachtsheim (2017) therefore recommend a DSD-specific method; this takes full advantage of the DSD's unique structure, and ensures that our assessment of MEs is independent of any second order effects!

- Their design-oriented approach is to split each "Y" response into orthogonal "YME" & "Y2nd" pseudo-responses, then apply a two-step method:
  - Step 1: Identify which factors are active (YME).
  - Step 2: If ≤3 active factors, assume hierarchy to limit the number of potential second order effects, then identify the best selection of hierarchical terms (Y2nd).

- **This "adaptive" approach ensures that we look at first order terms first, then delve further if (and only if) the data allows!**

Delicately Splitting our Data 1

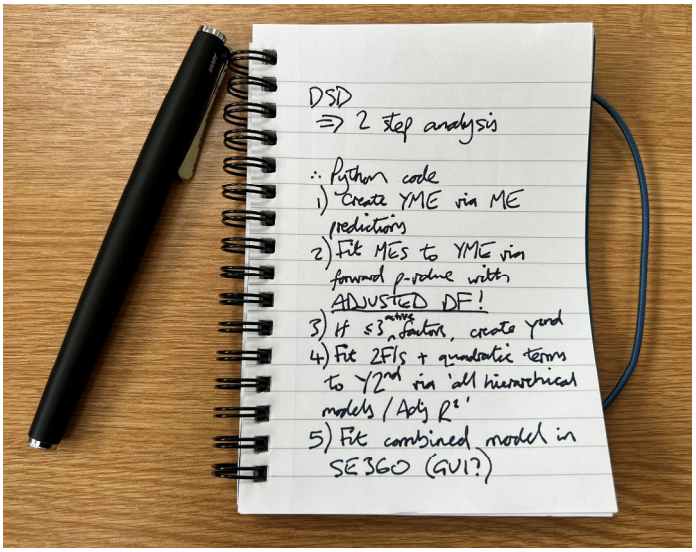
- A recommended approach is to split each Y response into orthogonal "YME" & "Y2nd" pseudo-responses…
- Step 1: Identify which factors are active (YME):
  - YME response is created by predicting Y values from Main Effects-only model
  - Then, use forward selection on p-values to identify active factors for YME
  - [Beware! Foldover design structure means that observations are not all truly independent, so fewer degrees of freedom (df) are available… so we must ensure that p-values are calculated using adjusted df!]
- If >3 active factors, then the supersaturated DSD does not have enough degrees of freedom to fit second order effects.
  - However, this is not a "failure": our efficient *2k+1* study will have revealed which factors merit further investigation!

Delicately Splitting our Data 2

- A recommended approach is to split each Y response into orthogonal "YME" & "Y2nd" pseudo-responses…
- Step 2: **If ≤3 active factors**, assume hierarchy to limit the number of potential second order effects, then identify the best selection (Y2nd):
  - Y2nd = Y – YME (i.e. residual from YME prediction, ensuring orthogonality to Main Effects)
  - By only considering the second-order effects associated with the active factors, we have reduced the number of possible model terms to fit within our available df!
  - We can then try fitting all hierarchical models to Y2nd for our active factors' 2FIs and quadratic terms, using Adj R2 to identify the best selection*

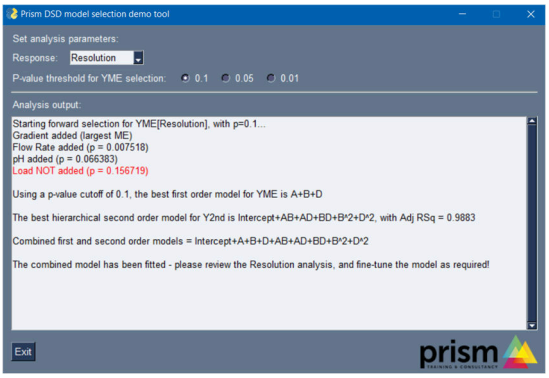
* Note: this criterion follows the method proposed in 2017; subsequently, the recommended selection method has evolved to use the ratio of Step 1 vs Step 2 RMSE values.

Daydreaming about Software Development



Diving into a Speedy Demo…

- Evolution of Python code:
  - v0: initial script, to establish SE connection and attempt basic interactivity.
  - v1: use Python *numpy*, *statsmodels* & *scipy* modules to perform forward p-value selection, with adjusted df for DSD.
  - v2: use *itertools* module to generate all subsets of potential 2nd order terms, then identify "best" model via Adj R2.
  - v3: create dialog via *PySimpleGUI* module, to obtain user inputs & present results; automatically fit the recommended model in SE360.



Done! Sayonara, Dudes!

- In summary:
  - Unique situations can require one-off solutions…
  - Routine, complex methods can be automated…
  - Users can be encouraged to apply the most appropriate analyses via custom tools…
  - **… all of which is now possible via Python in SE360!**

- With sincere thanks to:
  - My long-suffering colleagues at Prism;
  - Our good friends at Stat-Ease (especially Shari, for inviting us to talk today);
  - DoE practitioners, whose continued work inspires the ongoing evolution of DoE tools;
  - You, my Dear Statistical Devotees, for listening today!

- Any comments / questions? Feel free to get in touch via prismtc.co.uk/contact